THE FAST SKEW-CLOSURE ALGORITHM

Autor(en): **Fischer, M. J. / Paterson, M. S.**

Objekttyp: **Article**

Zeitschrift: **L'Enseignement Mathématique**

Band (Jahr): **26 (1980)**

Heft 1-2: **L'ENSEIGNEMENT MATHÉMATIQUE**

PDF erstellt am: **28.04.2024**

Persistenter Link: https://doi.org/10.5169/seals-51079

THE FAST SKEW-CLOSURE ALGORITHM [1]

by M. J. Fischer and M. S. Paterson

Abstract. A subtle matrix algorithm is explored and generalized. Originally used for transitive closures of symmetric Boolean matrices, this $O(n^2)$ algorithm computes a closure operation which is of interest for asymmetric and non-Boolean matrices too. The correctness of a generalized form of the algorithm is shown. The monoid generated by "skew-closure" and some of the more usual closures is investigated.

1. Introduction

The algorithm which forms the principal theme of this paper is of interest for several reasons. It is of mysterious ancestry; we have been unable to trace any published source which refers to it. It came to us by oral tradition at least seven years ago, when it impressed us with its speed and by the non-triviality of establishing its correctness. Further, whereas it seemed intended to be applied to Boolean matrices of symmetric and reflexive relations, the result of an application to more general matrices invited analysis.

The operation achieved by the algorithm we have termed "skew-closure". This closure is related to the more customary symmetric-and-transitive closure, and belongs to a very natural class of closure operations which we elaborate a little. In the interests of finding which matrix operations can be done equally rapidly, the monoid generated by several of these simple closures is treated in some detail. While this is finite, we later display a pair of slightly more complicated closures which together yield an infinite monoid.

[1] Presented at the *Symposium über Logik und Algorithmik* in honour of Ernst Specker, Zürich, February 1980.

2. The Boolean algorithm

We begin our presentation by giving the skew-closure algorithm in the simple form which is adequate for the Boolean case. Correctness results given here are easy corollaries of the more general theorem of a later section. The algorithm proceeds in an alternating series of passes: four passes in general, two in a special case.

A is an $n \times n$ Boolean matrix, and A_{i*} denotes the ith row of A. "$\vee$" represents disjunction and when applied to rows or matrices denotes a Boolean disjunction applied coordinate-wise. A partial order is defined by $A \geqslant B$ iff $A = A \vee B$.

The forward pass. For each row in turn, the leftmost non-zero entry to the right of the diagonal is sought. If found, the current row is "or"-ed into the row indexed by this entry's position. In an informal Algol this appears as:

```
for $i := 1$ step 1 until $n - 1$ do $\psi_i$
```

where ψ_i is

```
begin $k := i + 1$ step 1 until ($k = n$ or $A_{ik} \neq 0$)
    if $A_{ik} \neq 0$ then $A_{k*} := A_{k*} \vee A_{i*}$
end
```

The result is denoted by A^{Ψ}. The *backward pass*, resulting in $A^{\Psi'}$ is the same except that the iteration statements are

```
for $i := n$ step $-1$ until 2 do
```

and

```
$k := i - 1$ step $-1$ until ($k = 1$ ...
```

respectively. Thus it is the dual operation obtained by reversing the ordering of the rows and columns. One of these passes requires at most $O(n^2)$ operations on a random access machine. If a row operation on the matrix can be performed in a single step then only $O(n)$ of these are required and the time may be dominated by the searches for the first non-zero element after the diagonal in each row. This still uses $O(n^2)$ operations in a naive implementation but a more imaginative use of vector operations reduces this to at most $O(n \log n)$. In [1], we show a Turing machine implementation of the algorithm in time $O(n^2 \log n)$.

We denote the transpose of A by $\bar{A}$ and the reflexive-and-transitive closure of A by A^*. I is the unit matrix. The *skew-closure*, A^Q, of A is given by

$$A^Q = A \vee \bar{A}(\bar{A} \vee A)^* A$$

Further justification for this odd-looking operation will be given later, but for the present we have:

THEOREM 1.

(i) $A^Q = A^{\Psi\Psi'\Psi\Psi'}$,

(ii) *if* A *is reflexive, i.e.* $A \geqslant I$, *then* $A^Q = (\bar{A} \vee A)^* = A^{\Psi'\Psi\Psi'}$,

(iii) *if* A *is symmetric, i.e.* $A = \bar{A}$, *then*

$$A^Q = (\bar{A} \vee A)^+ = A^{\Psi\Psi'}.$$

Proof. Each result is a special case of the more general results in Theorem 6. □

An example where $A^{\Psi\Psi'\Psi} \neq A^{\Psi\Psi'\Psi\Psi'}$ is given by

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Both the (1, 2) and (2, 1) entries become 1 at the fourth pass.

If it appears to the reader that the choice of *earliest (latest)* non-zero entry in the forward (backward) pass algorithm is unnecessarily restrictive, she/he will be interested to know that with the modification to the forward pass of using

for $k := n$ step -1 until $(k = i+1$ ...

i.e. the *right*most non-zero to the right of the diagonal, and the corresponding change to the backward pass, the algorithm fails. Fortuitously, the same example as above serves. The (2, 1) entry remains zero after any number of passes.

The final pass of the algorithm can be regarded as copying the rows which have been built up, back into previous rows. The closure algorithm for reflexive symmetric matrices, in the form in which it was originally introduced to us, makes this explicit. It uses a single combined pass

```
for $i$ : = 1 step 1 until $n$ do
    begin $k$ : = $i + 1$ step 1 until ($k = n$ or $A_{ik} = 1$)
        if $A_{ik} = 1$ then $A_{k*}$ : = $A_{k*} \vee A_{i*}$
        else for $m$ : = 1 step 1 until $i - 1$ do
            if $A_{im} = 1$ then $A_{m*}$ : = $A_{i*}$
    end
```

It is not obvious that this algorithm has such a low time complexity, since it appears that the row copying step may be performed $O(n^2)$ times. However when the correctness of the algorithm is understood it becomes clear that each row is copied *into* at most once and so the total number of these operations is indeed $O(n)$.

We can give an informal proof using Theorem 1 that this algorithm is correct. We may think of A as representing an undirected graph on the index set $\{1, ..., n\}$. Since the algorithm causes no interaction between rows or columns corresponding to different components of the graph, it is sufficient to regard each component separately. We need only prove the correctness for a graph with a single component. It is plain that the nth row is the same after either the original algorithm or after $^{\Psi\Psi'}$. By Theorem 1, this must be all 1's provided that $n > 1$. But the copying operation of the original algorithm must have copied 1's throughout the entire matrix. This is correct.

We shall consider only our refined algorithm in further detail since it has natural generalizations which the old algorithm does not possess.

3. Basic closures

A matrix, A, regarded as a relation, is transitive if $A \geqslant A^2$. The transitive closure of A, A^T, is the least transitive matrix, X, containing A, and we may write

$$A^T = \mu X \,.\, X \geqslant A \vee X^2$$

(A^T is often denoted A^+). Similarly for the reflexive closure and symmetric closure

$$A^R = \mu X \,.\, X \geqslant A \vee I$$

$$A^S = \mu X \,.\, X \geqslant A \vee \overline{X}$$

We have also the reflexive-and-transitive closure

$$A^* = \mu X . X \geqslant A \vee I \vee X^2 .$$

Indeed for any formal polynomial P over $X, \bar{X}$, using product and disjunction with I as the identity, since $A \vee P(X, \bar{X})$ is monotonic in X, we have a unique minimal fixpoint $(\mu X . X \geqslant A \wedge P)$. Our interest in skew-closure is illuminated by the following result:

THEOREM 2.

$$A^Q = \mu X . X \geqslant A \vee \bar{X} X .$$

Proof. Firstly, A^Q satisfies the inclusion.

$$\begin{aligned} A \vee \overline{A^Q} A^Q &= A \vee (\bar{A} \vee \bar{A}(A \vee \bar{A})^* A)(A \vee \bar{A}(\bar{A} \vee A)^* A) \\ &\leqslant A \vee \bar{A}(\bar{A} \vee A)^* A = A^Q \end{aligned}$$

We note that $\bar{\bar{A}} = A$, $\overline{AB} = \bar{B}\bar{A}$ and $\overline{A^*} = \bar{A}^*$

Secondly, A^Q is minimal. Suppose $A^Q \underset{\neq}{>} K = (\mu X . X \geqslant A \vee \bar{X} X)$ and let m be the smallest integer such that $Q_m = A \vee \bar{A}(\bar{A} \vee A)^m A$ not $\leqslant K$. Obviously $A \leqslant K$, but also

$$\begin{aligned} \bar{A}(\bar{A} \wedge A)^m A &\leqslant \bigvee_{o \leqq r < m} \bar{A} A^r . \bar{A}(\bar{A} \vee A)^{m-r-1} A \vee \bar{A} A^m . A \\ &\leqslant \bigvee_{r<m} \overline{Q_r} . \bigvee_{r<m} Q_r \\ &\leqslant \bar{K} . K && \text{by minimality of } m \\ &\leqslant K && \text{by fixpoint property of } K \end{aligned}$$

This contradiction proves the Theorem. □

There are just two other monomials in $X, \bar{X}$ of degree at most two, namely $X\bar{X}$ and $\bar{X}\bar{X}$. The first yields a closure, Q', which is merely dual to skew-closure. The second yields a rather curious closure, T', which can be represented by the set of products over $A, \bar{A}$, defined by the strings

$$\{w \in \{A, \bar{A}\}^* \mid \text{number of } A\text{'s} \equiv 1 + \text{number of } \bar{A}\text{'s mod } 3\} \neg A(\bar{A}A)^+$$

Since the set of products defining T' is a regular set, this closure is computable using some fixed number of products, transitive closures, disjunctions and transposes. Therefore its computational complexity (like that of product [2]) is no greater than that of T, to within a constant factor. However we have been unable to show the converse.

Open Problem 1. Is there an $O(n^2)$ matrix-based algorithm for the T'-closure?

3. The quadratic monoid

To satisfy our curiosity we investigated the monoid generated by the composition of closures corresponding to polynomials of degree at most two. For any set of transformations E let M_E be the monoid generated by compositions of elements of E. For any polynomial $P(X, \bar{X})$, define
$Z_P : A \to (\mu X. X \geqslant A \vee P(X, \bar{X}))$
and then

$$\Pi_r = \{ Z_P \mid \deg(P) \leqslant r \}.$$

Theorem 3. $M_{\Pi_2} = M_{\{R, S, Q, Q', T, T'\}}$ *and the monoid is finite.*

Proof. The equality follows from the finiteness since

$$Z_{P_1 \vee P_2} = \bigvee_m \left(Z_{P_1} \cdot Z_{P_2}\right)^m$$

$$\in M_{\{Z_{P_1}, Z_{P_2}\}} \qquad \textit{if } \text{this is finite.}$$

$M_{\{R, S, Q, Q', T, T'\}}$ is examined explicitly below and is found to contain exactly fifty elements. □

We write A for the monoid identity given by $A^A = A$ and $[Z_1, \ldots, Z_k]$ for the closure $\bigvee_m (Z_1 \vee \ldots \vee Z_k)^m$. Together with the obvious idempotencies of closures we have the following sufficient defining relations.

$$W \stackrel{\text{def}}{=} [S, Q, Q', T, T']$$
$$= QQ' = Q'Q = QT' = Q'T' = SQ = SQ' = ST = ST'$$
$$V \stackrel{\text{def}}{=} [R, S, Q, Q', T, T'] = WR = RQ = RQ' = RT'$$
$$QT = [Q, T] \qquad Q'T = [Q', T]$$
$$T'Q = T'TQ = T'QT \qquad T'Q' = T'TQ' = T'Q'T$$
$$TT' = T'T \quad * \stackrel{\text{def}}{=} RT = TR \qquad RS = SR$$

The closures in the monoid are

$V \quad : \quad A^V = (\bar{A} \vee A)^*$

$W \quad : \quad A^W = (\bar{A} \vee A)^+$

$[Q, T]$: $A^{[Q,T]} = A^V . A$

$[Q', T]$: $A^{[Q',T]} = A . A^V$

$[T, T']$: $A^{[T,T']} = A \vee A^V . (AA \vee \bar{A}\bar{A}) . A^V$

$*, [R, S], R, S, Q, Q', T, T'$ and Λ .

The monoid can be counted after expressing its elements in a canonical form by the following rules.

(i) Using $RS = SR$, $RT = TR$, $RQ = RQ' = RT' = QQ'R$, we can bring any occurrence of R to the end of the product

(ii) Using $SQ = SQ' = ST = ST' = QQ'$, we can assume that any S occurs at the end of the rest of the product

(iii) $QT' = Q'T' = T'QQ'$ and $TT' = T'T$ allow us to bring any T' to the front of the remainder.

(iv) The elements generated by Q, Q', T are found to be

$$\Lambda, Q, Q', T, TQ, TQ', QT, Q'T, W$$

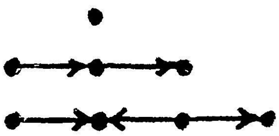
Prefixing these with T' yields only 4 new elements

$$T', T'Q, T'Q', T'T$$

(v) The 50 elements of the monoid are given by

$$\{\Lambda, Q, Q', T, T', TQ, TQ', QT, Q'T, T'Q, T'Q', T'T\} . \{\Lambda, R, S, SR\} \cup \{W, V\}$$

These elements are distinguishable by their effect on the graph



The "fast" monoid generated by the $O(n^2)$ operations R, S, Q, Q' has only 14 elements

$$\{\Lambda, Q, Q'\} . \{\Lambda, R, S, SR\} \cup \{W, V\}$$

Of computational interest are the relations

$$RQ = V \quad \text{and} \quad QQ' = SQ = W$$

which yield efficient ways to compute these common closures. Note that in some contexts the Q' closure may be more rapid to compute than S.

We illustrate some of the proofs for the results above.

THEOREM 4.

CANCELLATION LEMMA. *For all* A, $A\bar{A}A \geqslant A$.

Proof. $(A\bar{A}A)_{ij} \geqslant A_{ij}\bar{A}_{ji}A_{ij} = A_{ij}A_{ij}A_{ij} \geqslant A_{ij}$ □

(i) $RQ = V$

(ii) $QQ' = W$

(iii) $[Q, T] = QT$ and $A^{QT} = A^V . A$

Proof.

(i) If we show that $RQ \geqslant S$ the result follows easily. But

$$A^{RQ} \geqslant (I \vee A)^Q \geqslant A \vee \bar{A}I = A \vee \bar{A} = A^S$$

(ii) Again the only non-trivial part is that $QQ' \geqslant S$

$$A^{QQ'} \geqslant (A \vee \bar{A}A)^{Q'} \geqslant A \vee \bar{A}A . \bar{A} \geqslant A^S$$

by the Cancellation Lemma.

(iii) By inspection, $A^{[Q, T]} \leqslant A^V . A$

However,

$$A^V . A = A^* . \bar{A}A^VA \vee A^* . A \leqslant (A^Q)^T \leqslant A^{[Q, T]} \qquad \square$$

One of the harder results to prove is that $TT' = T'T$. We leave it as an exercise for the reader.

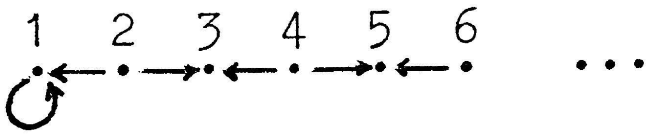
We have found that each mapping in Π_2 is defined by a regular set over $\{A, \bar{A}\}$, however in Π_3 there are «non-regular» mappings, e.g. $Z_{XX\bar{X} \vee \bar{X}XX}$.

The finiteness of M_{Π_2} does not persist for large r. We show by the example below that M_{Π_4} is infinite.

Open Problem 2. Is M_{Π_3} finite?

Example. Let $J = Z_{\bar{X}\bar{X}XX}$, $K = Z_{XX\bar{X}\bar{X}}$

$Z_{\bar{X}\bar{X}XX \vee XX\bar{X}\bar{X}} \notin M_{\{J,K\}}$ and so $M_{\{J,K\}}$ is infinite, since for the infinite graph shown below:



$(JK)^m$ adds all edges $< i, j >$ with $i, j \leqslant 2m$

and $(JK)^m J$ adds all edges $< i, j >$ with $i, j \leqslant 2m + 1$

Therefore J, JK, JKJ, ... are all distinct.

4. Generalized algorithm for power-group algebras

To elucidate the correctness of the algorithm and to encompass some more general applications we need to generalize from the $\{0, 1\}$ Boolean algebra to a slightly richer structure. The *power-group algebra* $P(G)$ is a structure defined from an arbitrary group G. The elements of $P(G)$ are the subsets of G; the operations we require are *union* ($\cup$), complex *product*:

$$ab = \{ gh \mid g \in a, \ h \in b \} \quad \text{for} \quad a, b \subseteq G$$

and *converse*:

$$\bar{a} = \{ g^{-1} \mid g \in a \}$$

$P(G)$ is a monoid with respect to product with identity $\lambda = \{\text{identity}_G\}$. As before we shall be considering matrices over the structure, with matrix product and union defined in the obvious way from product and union in $P(G)$, and matrix *converse* defined by

$$(\bar{A})_{ij} = \overline{A_{ji}}$$

The key properties of power-group algebras which are needed are given below

Lemma. *Let* a, b *be elements and* A, B *matrices*

(i) $\bar{\bar{a}} = a$; $\bar{\bar{A}} = A$

(ii) $\overline{ab} = \bar{b}\bar{a}$; $\overline{AB} = \bar{B}\bar{A}$

(*iii*) *if* $a \neq \varnothing$ *then* $a\bar{a} \supseteq \lambda$; $A\bar{A}A \supseteq A$

Proof. We prove only (iii). The first part is immediate and has the consequence that $a\bar{a}a \supseteq a$ for *all* a. For the second part

$$(A\bar{A}A)_{ij} \supseteq A_{ij}\,\bar{A}_{ji}\,A_{ij} = A_{ij}\,\overline{A_{ij}}\,A_{ij} \supseteq A_{ij}\,. \qquad \square$$

We observe that the $\{0, 1\}$ Boolean algebra is the power-group algebra corresppnding to the trivial one-element group. Other groups we shall use are $< \mathbf{Z}_k, + >$ and $< \mathbf{R}, + >$, the integers modulo k and the reals.

The operators $*$, $+$, V and W are defined just as before for matrices and elements. In the Boolean case we had the trivial results

$$\bar{a} = a^{+} = a^{W} = a$$

and

$$a^{*} = a^{V} = 1$$

In the general case we must augment the algorithm a little. Suppose for example there are edges labelled a, b from i to j and k respectively, and a self-loop at i labelled c. Then the label of the edge from j to k must eventually receive a term corresponding to the indirect paths from j to k i.e.

$$\bar{a}(a\bar{a} \vee c \vee \bar{c} \vee b\bar{b})^{*}\, b$$

The generalized form of ψ_i is:

```
$A_{i*} := A_{ii}{}^{V} A_{i*}$
$k := i + 1$ step 1 until ($k = n$ or $A_{ik} \neq \varnothing$)
if $A_{ik} \neq \varnothing$ then $A_{k*} := A_{k*} \cup \overline{A_{ik}}\, A_{i*}$
```

The programs for ψ_i', Ψ and Ψ' are analogous. They simplify to the programs of section 2 in the Boolean case.

The question of generalization could have been tackled axiomatically. Suppose $i < j < k$ and A_{ij}, $A_{ik} \neq \varnothing$. If row i were sent directly to row k we would have $\overline{A_{ik}}\, A_{i*}$, whereas via j we get

$$\overline{\overline{A_{ij}}\, A_{ik}} \,.\, \overline{A_{ij}}\, A_{i*} = \overline{A_{ik}}\, A_{ij}\, \overline{A_{ij}}\, A_{i*}$$

and we seem to require $A_{ij}\,\overline{A_{ij}} \geqslant \lambda$ for correctness. A power-group algebra seems the only structure of possible interest with this property.

Suppose that A is a matrix over $P(G)$ and we compute $A^{RQ} = A^{V}$. ${A^{V}}_{ij}$ is the set of elements of G which are the product of labels from a (weak) path from i to j in the graph corresponding to A. Each diagonal element

$A^V{}_{ii}$ is a subgroup of G and two diagonal elements in the same (weak) component are conjugate since we may show:

$$A^V{}_{ii} \cdot g = g \,.\, A^V{}_{jj} = A^V{}_{ij} \quad \text{for any} \quad g \in A^V{}_{ij}$$

In particular if all circuits on non-empty edges correspond to the group identity then in A^V each entry has at most one element. For example given a matrix A over $< \mathbf{R}, + >$ we determine from A^V for each component of the graph whether there are (weak) circuits with non-zero sums. If all circuits sum to zero then the graph is a "potential" graph, i.e. there is a function pot : $\mathbf{R} \rightarrow \{\text{vertices}\}$ such that each edge $< u, v >$ has the value (pot (v) − pot (u)). Similarly over $< \mathbf{Z}_k, + >$ we may determine whether each weak circuit of a directed graph has zero sum where forward and backward edges are accounted $+1$ and -1 respectively. Naturally we may find it convenient in some cases to hold only a homomorphic image of $P(G)$ for the computations e.g.

$$\begin{aligned} h(\varnothing) &= \varnothing \\ h(\{g\}) &= g \\ h(a) &= \omega \quad \text{when} \quad |a| > 1\,. \end{aligned}$$

5. Proofs of correctness

An operator ϕ is *monotonic* if $A \subseteq B$ implies $A^{\phi} \subseteq B^{\phi}$. ψ_i is *not* a monotonic operator but rather surprisingly Ψ *is*. To simplify our proofs we introduce several monotonic operators. Define ϕ_i by the program

```
$A_{i*} := A_{ii}^{V} A_{i*}$
for $k := i + 1$ step 1 until $n$ do $A_{k*} := A_{k*} \cup \overline{A_{ik}}\, A_{i*}$
```

and ϕ_i' analogously using "$i - 1$ step -1 until 1". Both are obviously monotonic. Φ and Φ' are defined from ϕ_i and ϕ_i' in a similar way to Ψ and Ψ'. Although $\Psi \subseteq \Phi$ is evident, the following result is not.

Theorem 5. *$\Psi = \Phi$ (and $\Psi' = \Phi'$).*

Proof. Consider ψ_i applied to an arbitrary matrix A and suppose it selects the index k with $A_{ik} \neq \varnothing$. (If no index is selected then of course $A^{\psi_i} = A^{\phi_i}$). We verify that:

(i) $A^{\phi_i\ \phi_{i+1} \cdots \phi_{k-1}} = A^{\phi_{i+1} \cdots \phi_{k-1}\ \phi_i}$ and similarly for ψ_i in place of ϕ_i

and (ii) $A^{\psi_i\ \phi_k} = A^{\phi_i\ \phi_k}$

(i) is immediate since ϕ_i and ψ_i do not affect any rows with indices between i and k.
To verify (ii) we check that $A^{\psi_i\ \phi_k} \supseteq A^{\phi_i}$ since for $j > k$

$$\begin{aligned}(A^{\psi_i\ \phi_k})_{j*} &\supseteq \overline{\overline{A_{ik}}\,A_{ij}}\,\overline{A_{ik}}\,A_{ii}^{V}\,A_{i*} \\ &\supseteq \overline{A_{ij}}\,A_{ii}^{V}\,A_{i*}\end{aligned}$$

We also note that $\phi_k\,\phi_k = \phi_k$
The proof of the Theorem is by induction on i from n to 1 for the equation

$$\phi_i \cdots \phi_n = \psi_i \cdots \psi_n$$

This is trivial for $i = n$, while for $i = 1$ it is the result to be proved. Suppose the equation true for $i + 1$, and then for an arbitrary A:

$$A^{\psi_i\,\psi_{i+1} \cdots \psi_n} = A^{\psi_i\,\phi_{i+1} \cdots \phi_n} \quad \text{by inductive hypothesis}$$

Either $A^{\psi_i} = A^{\phi_i}$ and we are done or $\exists\, k > i$ which is selected in ψ_i on A. Then

$$\begin{aligned}A^{\psi_i\ \phi_{i+1} \cdots \phi_k} &= A^{\phi_{i+1} \cdots \phi_{k-1}\,\psi_i\ \phi_k} && \text{by (i)} \\ &= A^{\phi_{i+1} \cdots \phi_{k-1}\,\phi_i\ \phi_k} && \text{by (ii)} \\ &= A^{\phi_i \cdots \phi_k} && \text{by (i)}\end{aligned}$$

The induction step now follows easily. □

In the proof of the Main Theorem below we need the following results.

LEMMA 1.

(i) $\Phi\Phi = \Phi$ *(and* $\Phi'\Phi' = \Phi'$*)*

(ii) $A^{\Phi\Phi'} \supseteq \bar{A}A \vee A$ *(and* $A^{\Phi'\Phi} \supseteq \bar{A}A \vee A$*)*

(iii) $A^{\Phi\Phi'\Phi} \supseteq \bar{A}\bar{A}A \vee \bar{A}A \vee A$

Proof.

(i) We may verify directly that for $i \leqslant j$, $\phi_j\,\phi_i \subseteq \phi_i\,\phi_j$

Then

$$\begin{aligned}\Phi\Phi &= \phi_1 \dots \phi_n \phi_1 \dots \phi_n \\ &\subseteq \phi_1\phi_1\phi_2\phi_2 \dots \phi_n\phi_n \quad \text{by repeated application of above inclusion} \\ &= \Phi \subseteq \Phi\Phi\end{aligned}$$

(ii) Consider an arbitrary contribution $\overline{A_{ik}}\, A_{ij}$ to $\bar{A}A$.

If $k > i$ then $\overline{A_{ik}}\, A_{ij} \subseteq A^{\phi_i} \subseteq A^{\Phi}$

else $\overline{A_{ik}}\, A_{ij} \subseteq A^{\phi'_i} \subseteq A^{\Phi'}$

(iii) $$\begin{aligned}A^{\Phi\Phi'\Phi} = (A^{\Phi\Phi'})^{\Phi'\Phi} &\supseteq (\bar{A}A \vee A)^{\Phi'\Phi} \\ &\supseteq (\bar{A}A \vee \bar{A})(\bar{A}A \vee A) \\ &\supseteq \bar{A}\bar{A}A \vee \bar{A}A\end{aligned}$$ □

Lemma 2. If $B \supseteq \bar{A}\bar{A}A \cup \bar{A}A \cup A$ and U is defined by

$$\begin{aligned}U_{ij} &= B_{ij} \quad \text{if} \quad i \leqslant j \\ &= \varnothing \quad \text{i|} \quad i > j\end{aligned}$$

then $\bar{B}\, U^V B \supseteq \bar{A}A^V A$

Proof.

If $j \leqslant k$, $A_{ij} A_{jk} \subseteq A_{ij} B_{jk} \subseteq A_{ij} U_{jk}$

If $j \geqslant k$, $A_{ij} A_{jk} \subseteq A_{ij} \bar{A}_{ji} A_{ij} A_{jk} \subseteq A_{ij} \bar{B}_{jk} \subseteq A_{ij} \bar{U}_{jk}$

Thus $AA \subseteq A\,(U \cup \bar{U})$. Similarly $\bar{A}\bar{A} \subseteq (U \cup \bar{U})\,\bar{A}$

Also $\bar{A}A \subseteq B$ and $\bar{A}A \subseteq \bar{B}$, so that $\bar{A}A \subseteq (U \cup \bar{U})$

From these inclusions we may derive

$$\begin{aligned}\bar{A}^+ . A^+ &\subseteq (U \cup \bar{U})^+ \\ \bar{A}^+ . A^+ &\subseteq \bar{B}\, U^V \\ \bar{A}^+ . A^+ &\subseteq U^V B \\ \bar{A}^+ . A^+ &\subseteq \bar{B}\, U^V B\end{aligned}$$

and finally

$$\bar{A}A^V A = (\bar{A}^+ A^+)^+ \subseteq \bar{B}\, U^V B$$ □

We shall consider weak paths which start with a backward edge, end with a forward edge and contain only those edges $< i, j >$ with $r \leqslant i \leqslant j$ for some threshold r. Hence we define the operators π_r for $1 \leqslant r \leqslant n + 1$.

$$X^{\pi_r} = \overline{X}\,(U^{(r)})^V X \cup X$$

$$\begin{aligned} \text{where } U^{(r)}{}_{ij} &= X_{ij} \text{ if } r \leqslant i \leqslant j \\ &= \varnothing \text{ otherwise} \end{aligned}$$

LEMMA 3.

$$\pi_r \subseteq \phi_r \pi_{r+1} \qquad \textit{for } 1 \leqslant r \leqslant n\,.$$

Proof. Let X be an arbitrary matrix, with $U^{(r)}$ defined as above and

$$Y = U^{(r)} \cup \overline{U^{(r)}}. \text{ Let } Z = X^{\phi_r}.$$

$$Y_{jr}\, Y^*_{rr}\, Y_{rk} \subseteq \overline{X_{rj}}\, X_{r\,r}^{\,V} X_{rk} \subseteq Z_{jk}$$

and likewise

$$Y_{jr}\, Y^*_{r\,r}\, Y_{rk} \subseteq \overline{Z_{jk}}$$

Similarly to deal with the ends of the paths,

$$\begin{array}{lll} \overline{X_{r*}}\, Y_{rr}{}^*\, Y_{rk} \subseteq \overline{Z_{k*}} & \text{if} & r < k \\ Y_{jr}\, Y_{rr}{}^*\, X_{r*} \subseteq Z_{j*} & \text{if} & r < i \\ \overline{X_{r*}}\, Y_{rr}{}^*\, X_{r*} \subseteq \overline{Z_{r*}}\, Z_{r*} & & \text{in any case} \end{array}$$

These inequalities show that internal edges of a path which visit vertex r can be replaced, so that $Z^{\pi_{r+1}}$ is sufficient. □

The effort is now behind us and the Main Theorem comes easily.

THEOREM 6.

(i) $Q = \Psi\Psi'\Psi\Psi'$

(ii) $V = R\Psi'\Psi\Psi'$

(iii) $W = S\Psi\Psi'$

Proof. The only matters requiring detailed proof are that the righthand transforms include Q. Let A be an arbitrary matrix.

For (i), define $B = A^{\Psi\Psi'\Psi}$

$$
\begin{aligned}
&= A^{\Phi\Phi'\Phi} && \text{by Theorem 5}\\
&\supseteq \bar{A}\bar{A}A \cup \bar{A}A \cup A && \text{by Lemma 1 (iii)}
\end{aligned}
$$

Therefore $\quad A^{Q} \subseteq B^{\pi_1} \quad$ from Lemma 2

For (ii) and (iii), let $B = A^{S}$

$$
\begin{aligned}
A^{R\Psi'\Psi} &= (I \cup A)^{\Phi'\Phi} && \text{by Theorem 5}\\
&\supseteq (I \cup \bar{A})(I \cup A) && \text{by Lemma 1 (ii)}\\
&\supseteq A \cup \bar{A}\\
&= B
\end{aligned}
$$

Also in this case, $A^{Q} \subseteq B^{\pi_1}$
In view of Theorem 5 and Lemma 1 (i) we have only to show now that $B^{\pi_1} \subseteq B^{\Phi\Phi'}$ to complete the proof. Using Lemma 3 repeatedly:

$$\pi_1 \subseteq \phi_1\pi_2 \subseteq \phi_1\phi_2\pi_3 \subseteq \ldots \subseteq \Phi\pi_{n+1}$$

But

$$X^{\pi_{n+1}} = \bar{X}X \cup X \subseteq X^{\Phi\Phi'}$$

therefore

$$\pi_1 \subseteq \Phi\Phi\Phi' = \Phi\Phi' \qquad \square$$

6. Conclusion

The close examination of a simple, practical matrix algorithm has led us to novel theoretical questions and to potentially useful generalizations of the algorithm. The principal contribution of this work to the programmer is the introduction of several very fast closure algorithms and the establishment of their correctness. The problems we have encountered in the theory of relations and closure operations have whetted our curiosity and suggest that further investigation may be rewarding.

Acknowledgment. We wish to thank Richard Ladner for discussions concerning this paper and its relation to security problems in protection systems.

REFERENCES

[1] Fischer, M. J., M. S. Paterson and N. Pippenger. *The Mailcarrier Problem* (in preparation).

[2] Fischer, M. J. and A. R. Meyer. Boolean matrix multiplication and transitive closure. *Conf. Record 1971 IEEE Annual Symposium on Theory of Computing*, 129-131.

(Reçu le 15 juillet 1980)

M. J. Fischer

Department of Computer Science
University of Washington
Seattle, Wa. 98195,
USA

M. S. Paterson

Department of Computer Science
University of Warwick
Coventry, CV4 7AL,
England

vide-leer-empty