The University of Warwick

THEORY OF COMPUTATION REPORT

NO. 10

ALGEBRAIC STRUCTURES FOR TRANSITIVE CLOSURE

by

DANIEL J. LEHMANN

Department of Computer Science
University of Warwick
COVENTRY
CV4 7AL
ENGLAND

FEBRUARY 1976

Algebraic structures for transitive closure

Abstract

Closed semi-rings and the closure of matrices over closed semi-rings are defined and studied.

Closed semi-rings are structures weaker than the structures studied by Conway [3] and Aho, Hopcroft and Ullman [1].

Examples of closed semi-rings and closure operations are given, including the case of semi-rings on which the closure of an element is not always defined.

Two algorithms are proved to compute the closure of a matrix over any closed semi-ring; the first one based on Gauss-Jordan elimination is a generalization of algorithms by Warshall, Floyd and Kleene; the second one based on Gauss elimination has been studied by Tarjan [11] and [12], from the complexity point of view in a slightly different framework.

Simple semi-rings, where the closure operation for elements is trivial, are defined and it is shown that the closure of an n×n-matrix over a simple semi-ring is the sum of its powers of degree less than n.

Dijkstra semi-rings are defined and it is shown that the rows of the closure of a matrix over a Dijkstra semi-ring, can be computed by a generalized version of Dijkstra's algorithm.

1. Introduction

Warshall's algorithm for computing the transitive closure of a Boolean matrix, Floyd's algorithm for minimum-cost paths, Kleene's proof that every regular language can be defined by a regular expression and Gauss-Jordan's method for inverting real matrices are different interpretations of the same program scheme (with one counter and an array).[1]

By program scheme is meant a terminating program with fixed control but where the sets over which the variables (or some of them) take their values and the meaning of the algebraic operations is left uninterpreted.

The purpose of this paper is to investigate the conditions of correctness for three such schemes for closure of matrices and to show a number of different structures in which they can be usefully applied.

The proof of correctness will be of algebraic type and under assumptions weaker than those made in previous works ([1], [2], [3]), and without introducing infinite sums.

The feeling that the numerical problem of inverting real matrices was closely related to some paths problems in graphs, has been part of the folkore of the subject for some time and has been recently expressed by Gondran [6], Backhouse and Carré [2] and Tarjan [11]; this work shows that, in a precise sense, both problems are special cases of the same general problem and proposes general algorithms which, when specialized, reduce to the methods mentioned above.

---

1 It was pointed out to the author by an anonymous referee that the algorithm for computing the transitive closure of a boolean matrix, generally attributed to Warshall, had been previously described by B. Roy [10].

The main novelty of this work is the definition of the closure of matrix by induction on the size of the matrix using a decomposition into sub-matrices. It is shown that such a definition implies the classical equation $A^* = I + A.A^*$ (1). In structures where (1) has more than one solution it is the author's experience that it is always a simple task to show equivalence of the inductive definition used and of any other reasonable definition, for example by means of least solutions to (1), when a suitable order can be defined.

2. Closed semi-rings

We shall consider algebras of the type $\{S, +, ., *, 0, 1\}$ where S is a set, $+: S\times S \to S$ and $.: S\times S \to S$ are binary operations, $*: S \to S$ is a unary operation, and $0 \in S$ $1 \in S$ are constants.
+ will be called addition, . multiplication and * closure.
In writing expressions we shall choose the infix notation a+b for +(a,b), a.b for .(a,b) and a^* for *(a)), assume that closure has precedence over the other operations and multiplication over addition.
Sometimes we shall also abbreviate a.b to ab.

Definition: An algebra is called a closed semi-ring iff the following equalities are identically true:

a) $a+(b+c) = (a+b)+c$ — addition is associative

b) $a+b = b+a$ — addition is commutative

c) $a+0 = a$ — 0 is a unit for addition

d) $a.(b.c) = (a.b).c$ — multiplication is associative

e) $a.1 = 1.a = a$ — 1 is a unit for multiplication

f) $a.(b+c) = a.b+a.c$
$(b+c).a = b.a+c.a$ — multiplication distributes over addition

g) $a^* = 1+a.a^* = 1+a^*.a$

Note: We do not ask for commutativity of multiplication, for idempotency of addition $(a+a = a)$,

for $(a+b)^* = (a^*b)^*b^*$, $(a.b)^* = 1+a.(b.a)^*.b$

or even for $a.0 = 0.a = 0$

It seems that axioms c) and e) asserting the existence of units for addition and multiplication are not essential and could have been left out had we chosen to axiomatize transitive closure proper (as opposed to reflexive transitive closure) but the formulae would have been much longer. It seems though that in certain interesting applications there is no zero element (see [7] p.160 where zero is called one)

Matrix Operations

Operations similar to addition, multiplication and closure can be defined on n×n matrices over a closed semi-ring, that make this set nearly a closed semi-ring.

Let A and B be n×n matrices over a closed semi-ring S.

$$A = \left[a_{ij}\right]_{i,j\in[1:n]} \qquad B = \left[b_{ij}\right]_{i,j\in[1:n]}$$

Let us define

$$A + B = \left[a_{ij} + b_{ij}\right]_{i,j\in[1:n]}$$

$$A.B = \left[\sum_{k\in[1:n]} a_{ik}\, b_{kj}\right]_{i,j\in[1:n]}$$

The closure operation on matrices is defined inductively on the size of the matrix by decomposing the matrix into four sub-matrices. The definition is correct because, as will be shown in the next paragraph, the size of the sub-matrices used in this decomposition does not bear any relevance on the definition.

Definition of the closure of a n×n matrix:

If $n = 1$ $[a]^* = [a^*]$

If $n > 1$ and $A = \begin{bmatrix} B & C \\ D & E \end{bmatrix}$ where, for some $0 < k < n$:

$B : k \times k$, $C : k \times (n-k)$, $D : (n-k) \times k$, $E : (n-k) \times (n-k)$,

then

$$A^* = \begin{bmatrix} B^* + B^*C\Delta^*DB^* & B^*C\Delta^* \\ \Delta^*DB^* & \Delta^* \end{bmatrix} \qquad \text{for } \Delta = E + DB^*C$$

Note: It is not true in general that in this definition $B^* + B^*C\Delta^*DB^*$ can be replaced by $(B + CE^*D)^*$; however Conway [1] has shown that if three more identities are true in the closed semi-ring

$a.0 = 0.a = 0$, $(a.b)^* = 1 + a.(b.a)^*.b$ (which implies our g)) and $(a + b)^* = (a^*b)^*.a^*$, then the above replacement is possible and the corresponding identities for matrices hold. This is probably so even if only the last two identities are assumed.

Conversely it is easy to see that the validity of the above replacement implies the last two identities in the presence of the first one.

Let us now define two matrices of constants:

$$0_n = [c_{ij}]_{i,j\in[1:n]} \quad \text{with} \quad c_{ij} = 0 \quad \text{for } i,j\in[1:n]$$

$$I_n = [\delta_{ij}]_{i,j\in[1:n]} \quad \text{with} \quad \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

It is easy to verify that the analog of identities a), b), c), d) and f) hold for matrices.

Note: the analogue of e): $A.I_n = I_n.A = A$ does not hold.

Correctness of the inductive definition of closure

The proof that the size of the sub-matrices involved in the definition is irrelevant boils down to computing the closure of a matrix with nine sub-matrices in two different ways:

$$\left[\begin{array}{c|cc} A & B & C \\ \hline D & E & F \\ G & H & I \end{array}\right] \quad \text{and} \quad \left[\begin{array}{cc|c} A & B & C \\ D & E & F \\ \hline G & H & I \end{array}\right]$$

and verifying nine identities. The verification is trivial using commutativity and associativity of matrix addition, associativity of matrix multiplication and distributivity of matrix multiplication over matrix addition.

Note that axiom g) is not used in this proof.

The verification is carried out in Appendix 1 of [9].

Completion of a partial closed semi-ring

Define a partial closed semi-ring to be an algebra of the type described above, where closure is only a partial function and satisfying a) ... f) and g) whenever the closure of a is defined. If S is a partial closed semi-ring then $S \cup \{u\}$ (where $u \notin S$ is a new element and stands for undefined) can be made a closed semi-ring by adding these definitions: $u+a = a+u = u$, $a.u = u.a = u$, $u^* = u$ and $a^* = u$ if a^* was not previously defined.

$S \cup \{u\}$ is called the completion of S.

The careful reader now understands why the trouble was taken not to include the identity $a.0 = 0.a = 0$ in the list of axioms and to deal with identity matrices I_n which are not real identities.

We shall now prove that the analogue of axiom g) holds for matrices. But first a lemma:

Lemma 1: If A and B are n×n matrices over a closed semi-ring then:

1) $(I_n + B).A = A + B.A$

2) $A\,(I_n + B) = A + A.B.$

Proof:

$$\left[(I_n + B).A\right]_{i,j} = \sum_{k=1,n} (\delta_{ik} + b_{ik})a_{kj} = 1.a_{ij} + b_{ii}.a_{ij} + \sum_{\substack{k=1,n \\ k \neq i}} (0 + b_{ik})a_{kj}$$

$$= a_{ij} + \sum_{k=1,n} b_{ik}\, a_{kj}$$

And symmetrically for 2).

Theorem 1: If A is a n×n matrix then:

$$A^* = I_n + A.A^* = I_n + A^*.A$$

Proof: The two equalities being symmetric let us just prove the first one. By induction on n.

If $n = 1$ $a^* = 1 + a.a^*$ by g)

If $n > 1$ suppose $A = \begin{bmatrix} C & D \\ E & F \end{bmatrix}$ $C : k \times k$

With $\Delta = F + EC^*D$, by definition:

$$A^* = \begin{bmatrix} C^* + C^*D\Delta^*EC^* & C^*D\Delta^* \\ \Delta^*EC^* & \Delta^* \end{bmatrix}$$

$$A.A^* = \begin{bmatrix} CC^* + CC^*D\Delta^*EC^* + D\Delta^*EC^* & CC^*D\Delta^* + D\Delta^* \\ EC^* + EC^*D\Delta^*EC^* + F\Delta EC^* & EC^*D\Delta^* + F\Delta^* \end{bmatrix}$$

But by the induction hypothesis:

$$C^* = I_k + C.C^* \quad \text{and} \quad \Delta^* = I_{n-k} + \Delta\Delta^*$$

By lemma 1:

$$D\Delta^*EC^* + CC^*D\Delta^*EC^* = (I_k + CC^*)D\Delta^*EC^* = C^*D\Delta^*EC^*$$

$$D\Delta^* + CC^*D\Delta^* = (I_k + CC^*)D\Delta^* = C^*D\Delta^*$$

$$EC^* + EC^*D\Delta^*EC^* + F\Delta^*EC^* = EC^* + \Delta\Delta^*EC^* = (I_{n-k} + \Delta\Delta^*)EC^*$$
$$= \Delta^*EC^*$$

$$EC^*D\Delta^* + F\Delta^* = \Delta\Delta^*$$

Then $I_n + A.A^* = \begin{bmatrix} I_k + CC^* + C^*D\Delta^*EC^* & C^*D\Delta^* \\ \Delta^*EC & I_{n-k} + \Delta\Delta^* \end{bmatrix}$

$= A^*$ by the induction hypothesis.

Q.E.D.

Corollary 1: $A.A^* = A + A.A^*A = A^*A$

By Lemma 1 and Theorem 1

Corollary 2: $A^* = I_n + A + AA^*A$

By Theorem 1 and Corollary 1.

Corollary 3: $B + AA^*B = A^*B$ and $B + BA^*A = BA^*$

By Lemma 1 and Theorem 1.

---

3. Examples of closed semi-rings

Boolean semi-ring: $\{ \{0,1\}, \vee, \wedge, T, 0, 1 \}$ where $T(0) = T(1) = 1$

The closure of a Boolean matrix is its transitive and reflexive closure.

A proof of that fact can be obtained either directly by induction or using paragraph 5 on simple semi-rings.

$\{\mathbb{R}_+ \cup \{+\infty\}, \text{Min}, +, Z, +\infty, 0\}$ where $\mathbb{R}_+$ is the set of non-negative real numbers

is a closed semi-ring where $Z(a) = 0 \quad a \in R_+ \cup \{+\infty\}$.

The closure of a matrix over this semi-ring is the minimum-cost matrix for the labelled graph yielded by the matrix.

$\underline{\{\mathbb{R} \cup \{+\infty, -\infty\}, \text{Min}, +, *, +\infty, 0\}}$ where $\mathbb{R}$ is the set of real numbers and $a^* = \begin{cases} 0 & \text{if } a \geq 0 \\ -\infty & \text{if } a < 0 \end{cases}$

is a closed semi-ring, if $(+\infty) + (-\infty) = +\infty$.

The closure of a matrix gives the minimum-cost matrix for the corresponding labelled graph, or $-\infty$ when there are paths of cost as small as desired.

Similarly $\{Q \cup \{+\infty, -\infty\}, \text{Min}, +, *, +\infty, 0\}$

and $\{Z \cup \{+\infty, -\infty\}, \text{Min}, +, *, +\infty, 0\}$ are

closed semi-rings,

and so is

$$\{\mathbb{R}_+ \cup \{+\infty, -\infty\}, \text{Max}, +, \circledast, -\infty, 0\}$$

with $$a^{\circledast} = \begin{cases} +\infty & \text{if } a > 0 \\ 0 & \text{if } a = 0 \end{cases}$$

On this last closed semi-ring the closure of a matrix gives the maximum cost paths in the corresponding graphs or $+\infty$ if there are paths of unbounded cost.

$\underline{\{\mathbb{R}_+ \cup \{+\infty\}, \text{Max}, \text{Min}, \infty, 0, +\infty\}}$ where $\infty(a) = +\infty$ is a closed semi-ring and the closure of a matrix over this ring gives the maximum-capacity paths.

More generally if L is a lattice with operations $\vee$ and $\wedge$ and bottom $(\bot)$ and top $(\top)$ then $\{L, \vee, \wedge, T, \bot\ \top\}$ with $T(a) = \top$ is a closed semi-ring.

$\underline{\{P(\Sigma^*), \cup, . , *, \phi, \varepsilon\}}$ is a closed semi-ring if

Σ is an alphabet, ε the empty word, . concatenation and

$$A^* = \bigcup_{i \in N} A^i .$$

$\underline{\{\mathbb{R} \cup \{u\}, +, . , s, 0, 1\}}$ is a closed semi-ring for

$$s(a) = \frac{1}{1-a} \quad \text{for } a \neq 1 \quad \text{and} \quad 1^* = u$$

and $a + u = u + a = u$ $u.a = a.u = u$ and $u^* = u$.

The same is true if $\mathbb{R}$ is replaced by $\mathbb{C}$.

In this closed semi-ring, if a matrix A is such that A^* does not contain u then $A^* = (I-A)^{-1}$ (A does not contain u either), by theorem 1.

A^{-1} may be computed by computing the closure of I-A, at least if $(I-A)^*$ does not contain u.

Unfortunately there are non-singular matrices A such that $(I-A)^*$ does contain u.

Still if P is a permutation matrix such that $(I-PA)^*$ does not contain u then

$$(I-PA)^* = (PA)^{-1} = A^{-1} P^{-1} \quad \text{and}$$

$(I-PA)^*P = A^{-1}$, and the computation of A^{-1} may be reduced to that of closure.

Conversely, if A is non-singular there is a permutation matrix P such that PA can be inverted by Gaussian elimination without pivoting. As shall be seen later Gaussian elimination method without pivoting applied on B computes $(I-B)^*$, then $(I-PA)^* = (PA)^{-1}$ and does not contain u.

$\{F, \sqcup, \circ, *, \lambda x\bot, \lambda xx\}$ is a closed semi-ring if L is a complete lattice with zero element $\bot$ ($\bot \sqcup a = a$) and upper-operation $\sqcup$,

F is the set of all functions: $L \to L$ satisfying :

$$(2) \quad f(\sqcup A) = \sqcup\{f(a) \mid a \in A\} \text{ for any } A \subseteq L,\ A \neq \emptyset$$

$\sqcup$ is defined by $(f \sqcup g)(x) = f(x) \sqcup g(x)$ with an obvious notational ambiguity,

$\circ$ is function composition, $\lambda x\bot$ the constant function bottom and λxx the identity, and $*$ is defined by:

$$(3) \quad f^*(x) = \sqcup\{f^i(x) \mid i = 0, 1, \dots\} .$$

On this semi-ring the computation of the closure of a matrix amounts to a global data flow problem [7].

All distributive global data flow problems can be treated as transitive closure problems but non-distributive problems, where (2) is replaced by the weaker assumption that the functions of F are monotone, do not seem to fit into our framework.

4. Warshall-Floyd-Kleene's algorithm. Gauss-Jordan method.

An algorithm will now be presented, to compute the closure of a matrix.

<u>WFK- algorithm</u> ([10], [13], [5], [8])

Input: $A = \left[a_{ij}\right]_{i,j\in[1:n]}$ $a_{ij} \in S$ closed semi-ring

begin

1. **for** each $i,j\in[1:n]$ **do** $A_0[i,j] \leftarrow A[i,j]$;
2. **for** $k:=1$ **step** 1 **until** n **do**
3. **for** each $i,j\in[1:n]$ **do**
4. $A_k[i,j] \leftarrow A_{k-1}[i,j] + A_{k-1}[i,k].\left(A_{k-1}[k,k]\right)^* A_{k-1}[k,j]$;
5. **for** each $i,j\in[1:n]$ **do**
6. $R[i,j] \leftarrow \delta_{ij} + A_n[i,j]$;

end

Output: $R[i,j]$ for $i,j\in[1:n]$

<u>Note</u>: δ_{ij} in line 6 is 1 for $i = j$ and 0 otherwise.

This algorithm is a straightforward translation of Kleene's proof that every regular language can be represented by a regular expression. Floyd's algorithm for minimum-cost paths in directed graphs is a specialization of the above algorithm to the case where $a^* = 1 \; \forall a\in S$ and Warshall's algorithm for the transitive closure of Boolean matrices is its specialization to the closed semi-ring {0,1}.

The algorithm computes the "transitive" closure of A in A_n and its "transitive and reflexive" closure in B.

Its specialization to the closed semi-ring $\mathbb{R} \cup \{u\}$ is Gauss-Jordan method for inverting matrices, without pivoting.

The repetitive statements used are of two types, the for statement of ALGOL, and a for each statement indicating that the order in which the values are given is of no importance.

For each $i,j \in [1:n]$ is an abbreviation for

For each $(i,j) \in [1:n] \times [1:n]$.

The algorithm uses n+1 different matrices A_k $(0 \leq k \leq n)$ for simplicity. It is not difficult to write an equivalent algorithm using only one such matrix, taking care that entries in the matrix are not changed before they are used.

We shall now proceed to proving that WFK-algorithm computes in R the closure of the input matrix A.

Notations: If C is a n×n-matrix let us define $C_{[i,k][j,\ell]}$ to be its submatrix consisting of rows i to k and columns j to ℓ.

$$(1 \leq i \leq k \leq n \;,\; 1 \leq j \leq \ell \leq n).$$

To simplify this notation the full interval [1:n] will be abbreviated to . and the one element interval [i,i] to i.

Examples: $A_{i.}$ is the i^{th} row of A

A_{ij} is the element A[i,j]

In matrix notation the algorithm computes a sequence of n×n matrices $A^{(k)}$ for $0 \leq k \leq n$ defined by:

$$A^{(o)} = A$$

$$A^{(k)} = A^{(k-1)} + A^{(k-1)}_{.k} \cdot A^{(k-1)*}_{kk} \cdot A^{(k-1)}_{k.} \qquad \text{for } 1 \leq k \leq n$$

and the output R by:

$$R = I_n + A^{(n)}$$

We shall now prove that $A^{(n)} = A + A.A^*.A$, the proof not relying on assumption g).

Theorem 2: For any $k \in [o:n]$

$$A^{(k)} = A + A_{.[1:k]}\left(A_{[1:k][1:k]}\right)^* A_{[1:k]} \;.$$

(with the convention that $A_{.[1:0]}$, $A_{[1:0],[1:0]}$ and $A_{[1:0].}$ should just be ignored).

This obviously implies $A^{(n)} = A + A\,A^*A$.

Proof: By induction on k.

For k = 0 $A^{(o)} = A$.

For $k = \ell + 1 \qquad (0 \leqslant \ell \leqslant n-1)$

$$A^{(k)} = A^{(\ell)} + A^{(\ell)}_{.k}\,(A^{(\ell)}_{kk})^*\,A^{(\ell)}_{k.} \qquad (1)$$

by the preceding matrix-form of WFK-algorithm;
and by the induction hypothesis:

$$A^{(\ell)} = A + A_{.[1:\ell]}(A_{[1:\ell][1:\ell]})^*\,A_{[1:\ell].} \qquad (2)$$

Define $B = A_{[1:\ell][1:\ell]}$, $P = A_{k[1:\ell]}$, $Q = A_{[1:\ell]k}$.

Then:

$$A^{(\ell)}_{.k} = A_{.k} + A_{.[1:\ell]}B^*Q$$

$$A^{(\ell)}_{k.} = A_{k.} + PB^*A_{[1:\ell].}$$

$$A^{(\ell)}_{kk} = A_{kk} + PB^*Q$$

Define $\Delta = A^{(\ell)}_{kk} = A_{k,k} + PB^*Q$

The respective positions of B, P and Q in A are illustrated by Fig. 1

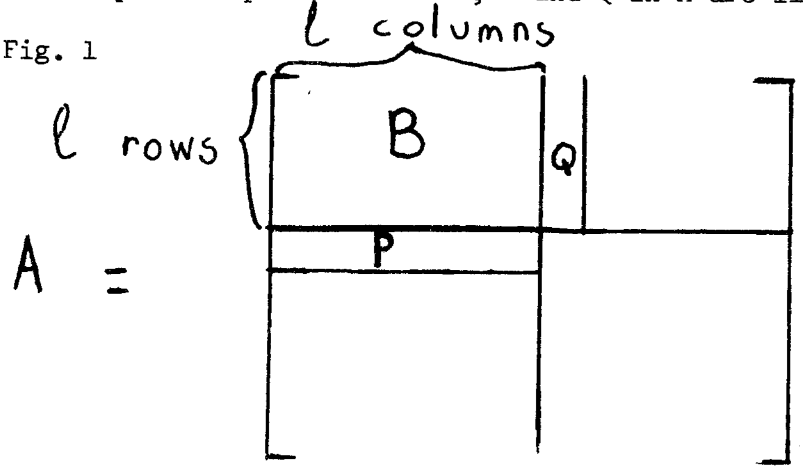

Fig. 1

Rewriting (1) using (2) we get:

$$A^{(k)} = A + A_{.[1:\ell]}B^*\,A_{[1:\ell].} + \left[A_{.k} + A_{.[1:\ell]}B^*Q\right]\,\Delta^*\,\left[A_{k.} + PB^*A_{[1:\ell].}\right] \qquad (3)$$

But, by definition of the closure operation:

$$\left(A_{[1:k][1:k]}\right)^* = \begin{bmatrix} B & Q \\ P & A_{kk} \end{bmatrix}^* = \begin{bmatrix} B^* + B^*Q\Delta^*PB^* & B^*Q\Delta^* \\ \Delta^*PB^* & \Delta^* \end{bmatrix}$$

and

$$A_{.[1:k]} \left(A_{[1:k][1:k]}\right)^* A_{[1:k].} =$$

$$A_{.[1:\ell]} (B^* + B^*Q\Delta^*PB^*) A_{[1:\ell].} + A_{.k} \Delta^*PB^*A_{[1:\ell].} + A_{.[1:\ell]}B^*Q\Delta^*A_{k.}$$
$$+ A_{.k} \Delta^*A_{k.}$$

Comparing with (3) gives $A^{(k)} = A + A_{.[1:k]} \left(A_{[1:k][1:k]}\right)^* A_{[1:k].}$

Q.E.D.

<u>Corollary</u> (using g) again): $R = A^*$

<u>Proof</u>: $R = I_n + A^{(n)} = I_n + A + AA^*A$ and by corollary 2 to Theorem 1:

$$R = A^*.$$

5. Gauss method

Another algorithm shall now be introduced for computing the closure of a matrix, the specialization of which to the semi-ring $\mathcal{R} \cup \{u\}$ is Gauss algorithm for inverting real matrices (without pivoting).

<u>Gauss algorithm</u>:

Input: $A = [a_{ij}] i,j \in [1:n]$ $a_{ij} \in S$ closed semi-ring

begin

1. **for** each $i,j \in [1:n]$ do $G_o[i,j] \leftarrow A[i,j]$;
2. **for** $k:=1$ **step** 1 **until** n **do**
3. **for** each $i,j \in [k:n] \times [1:n]$ **do**

4. $G_k[i,j] \leftarrow G_{k-1}[i,j] + G_{k-1}[i,k].\left(G_{k-1}[k,k]\right)^* G_{k-1}[k,j];$

5. **for** each $i,j \in [1:n]$ **do**

6. $\quad B[i,j] \leftarrow G_i[i,j];$

7. **for** $i:=n-1$ **step** -1 **until** 1 **do**

8. $\quad$ **for** each $j,k \in [1:n] \times [i+1:n]$ **do**

9. $\quad B[i,j] \leftarrow B[i,j] + G_i[i,k]\ B[k,j];$

10. **for** each $i,j \in [1:n]$ **do**

13. $\quad R'[i,j] \leftarrow \delta_{ij} + B[i,j];$

end

Output: $R'[i,j]$ for $i,j \in [1:n]$

Remarks: The algorithm is a straightforward translation of Gauss inversion method.

In the version presented above the use of memory space is very inefficient but, as with WFK-algorithm, it can be reduced to one n×n matrix by obvious changes.

The essential differences with WFK are that in statement 3 i runs only from k to n instead of from 1 to n and that a second pass, upwards, takes place after statement 7.

The advantage of Gauss method is apparent when one may suppose that 0.a = a.0 = 0 (for all the a's which arise during the execution of the algorithm) and when the input matrix A contains a large number of zeros. In this case the zeros stay in longer in Gauss method than in WFK.

In [11] Tarjan, under assumptions close to ours but seemingly incomparable with them, has shown that, with suitable data representation, Gauss method may be implemented in a number of basic steps (on a random access machine) which is almost linear in the number of non-zero entries in the input matrix for a large class of matrices with restricted zero-non-zero structure. The author is hopeful that this remains true under the present assumptions (when a.0 = a.0=0).

We shall now proceed to showing that the above algorithm computes in R' the closure of the input matrix A. References will be made to the notations used in the proof of correctness of WFK.

Clearly, in matrix notation, the first pass of the algorithm (statements 1-4) computes of sequence of n+1 matrices

$G^{(o)}, G^{(1)} \ldots G^{(n)}$ such that:

$G^{(o)} = A^{(o)} = A, \quad G^{(1)} = A^{(1)}$, and $G^{(k)} = A^{(k)}_{[k:n]}$ for $1 \leq k \leq n$.

Then in statements 5-6 it computes a matrix $B^{(o)}$ such that

$B^{(o)}_{k.} = A^{(k)}_{k.}$ for $1 \leq k \leq n$,

or more picturesquely

$$B^{(o)} = \begin{bmatrix} A^{(1)}_{1.} \\ A^{(2)}_{2.} \\ \vdots \\ A^{(n)}_{n.} \end{bmatrix}$$

Then in statements 7-9 the algorithm computes a sequence of row vectors $B^{(n)}, \ldots, B^{(1)}$ such that:

$B^{(n)} = B^{(o)}_{n.} = A^{(n)}_{n.}$ and

$$B^{(k)} = B^{(o)}_{k.} + B_{k[k+1:n]} \begin{bmatrix} B^{(k+1)} \\ B^{(k+2)} \\ \vdots \\ B^{(n)} \end{bmatrix}$$

<u>Theorem 3</u>: For any k $1 \leq k \leq n$, $B^{(k)} = A^{(n)}_{k.}$

<u>Proof</u>: By backwards induction on k

For $k = n$ $B^{(n)} = A^{(n)}_{n.}$

For $k = \ell-1$

$$B^{(k)} = B^{(o)}_{k.} + B^{(o)}_{k[k+1:n]} \begin{bmatrix} B^{(k+1)} \\ B^{(k+2)} \\ \vdots \\ B^{(n)} \end{bmatrix}$$

$= A^{(k)}_{k.} + A^{(k)}_{k[k+1:n]} \cdot A^{(n)}_{[k+1:n]}$. by the induction hypothesis.

Let us now consider a partition of A into sub-matrices

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix} \quad \text{such that B is k×k.}$$

Precisely: $B = A_{[1:k][k:k]}$, $C = A_{[1:k][k+1:n]}$,

$D = A_{[k+1:n][1:k]}$, $E = A_{[k+1:n][k+1:n]}$

By Theorem 2: $A^{(k)} = A + A_{\cdot[1:k]}\left(A_{[1:k][1:k]}\right)^{*} A_{[1:k]\cdot}$,

and $A^{(n)} = A + AA^*A$.

or using the partition into sub-matrices:

$$A^{(k)} = \begin{bmatrix} B & C \\ D & E \end{bmatrix} + \begin{bmatrix} B \\ D \end{bmatrix} B^* \; [B \; C]$$

$$= \begin{bmatrix} B + BB^*B & C + BB^*C \\ D + DB^*B & \Delta \end{bmatrix} \quad \text{if we define } \Delta = E + DB^*C \text{ .}$$

By Corollaries 1 and 3 to Theorem 1: $B + BB^*B = B^*B$ $A + AA^*A = A^*A$

$C + BB^*C = B^*C$ and $D + DB^*B = DB^*$.

Consequently:

$$A^{(k)} = \begin{bmatrix} B^*B & B^*C \\ DB^* & \Delta \end{bmatrix}, \text{ and}$$

$$A^{(n)} = A^*A = \begin{bmatrix} B^* + B^*C\Delta^*DB^* & B^*C\Delta^* \\ \Delta^*DB^* & \Delta^* \end{bmatrix} \begin{bmatrix} B & C \\ D & E \end{bmatrix}$$

because the definition of the closure of a matrix is independent of the size of the sub-matrices chosen.

Then

$$A^{(k)}_{k\cdot} = \begin{bmatrix} B^*_{k\cdot}B & B^*_{k\cdot}C \end{bmatrix}$$

$$A^{(k)}_{k[k+1:n]} = B^*_{k\cdot}C$$

$$A^{(n)}_{[k+1:n]\cdot} = [\Delta^*DB^*B + \Delta^*D \quad \Delta^*DB^*C + \Delta^*E] = [\Delta^*DB^* \quad \Delta^*\Delta]$$

Then:

$$B^{(k)} = A^{(k)}_{k.} + A^{(k)}_{k[k+1:n]} \quad A^{(n)}_{[k+1:n]}.$$

$$= \begin{bmatrix} B^*_{k.} B + B^*_{k.} C\Delta^* DB^* & B^*_{k.} C + B^*_{k.} C\Delta^*\Delta \end{bmatrix}$$

$$= \begin{bmatrix} B^*_{k.} B + B^*_{k.} C\Delta^* DB^* & B^*_{k.} C\Delta^* \end{bmatrix}$$

But

$$A^{(n)}_{k} = \begin{bmatrix} B^*_{k.} B + B^*_{k.} C\Delta^* DB^* B + B^*_{k.} C\Delta^* D & B^*_{k}.C + B^*_{k.} C\Delta^* DB^* C + B^*_{k.} C\Delta^* E \end{bmatrix}$$

$$= B^{(k)}$$

Q.E.D.

From that it follows that the output matrix R' is:

$$R' = I_n + \begin{bmatrix} B^{(1)} \\ \vdots \\ B^{(n)} \end{bmatrix} = I_n + A^{(n)} = A^* .$$

6. Simple semi-rings

A class of closed semi-rings will now be defined in which the star operation is simple to perform: $a^* = 1$ for any $a \in S$. A characterization of the closure of a matrix over a simple semi-ring will be given that relates the closure of a matrix to the sum of the labels of the elementary paths between couples of nodes.

Simple semi-rings are exactly the Q-semi-rings of Yoeli [14], and the fundamental property below shows that our definition of closure is a correct version of his not quite correct definition of the transmission matrix (not quite correct because infinite sums are used without ever being properly defined; similar carelessness is found in [1] and [2].).

The regular algebras of Carré and Backhouse are close to our simple semi-rings (they do not assume a+1 = 1 but assume a+a = a and a rule of inference); their axiomatization makes an extensive use of the order: $a \leq b$ iff a+b = b and this seems to take us far away from linear algebra. The author does not know how to compare the strength of the axioms for regular algebras and simple semi-rings.

Definition: A closed semi-ring is called simple iff, in addition to assumptions a)...g), the following is true

h) $a+1 = 1$.

A number of identities follow, for example: $1 + 1 = 1$,

$$a + a = a, \quad a^* = 1, \quad a + a.b = a + b.a = a,$$

$$a.b. + a.c.b = a.b, \quad 0.a = a.0 = 0.$$

The last of these identities is proved by: $0.a = 0 + 0.a = 0(1 + a) = 0.1 = 0.$

The next theorem will provide a link between closure of matrices and labelled paths in a graph and be used to prove as a Corollary that, over simple semi-rings, closure behaves reasonably with respect to interchanging at the same time rows and columns.

This last result has already been proved in [3] by Conway (p.111) under much weaker assumptions (though the whole proof has not been printed) and as it is the only result of importance for the next section, a reader familiar with Conway's results and uninterested in graphs may skip to the next section.

Fundamental property of simple semi-rings

If $A = \left[a_{ij}\right]_{i,j\in[1:n]}$ is a n×n matrix over a simple semi-ring and $B = A^* = b_{ij\ i,j\in[1:n]}$

then
$$b_{ij} = \delta_{ij} + \sum_{\substack{m \\ k_1,\ldots,k_m\in[1:n] \\ k_1,\ldots,k_m \text{ all distinct} \\ \text{and different from } i \text{ and } j.}} a_{ik_1}\, a_{k_1k_2} \cdots a_{k_{m-1}k_m}\, a_{k_mj}$$

A full proof is given in Appendix 2 of [9] and a brief summary will only be given here.

Sketch of the proof of the fundamental property of simple semi-rings

There is an obvious way to look at a nxn matrix as a labelled complete directed graph on n vertices, and to attach a label to all directed paths.

The fundamental property of simple semi-rings says that the (i,j) -th element of the closure of a matrix A is the sum of the labels of all elementary paths from i to j. The property can be proved by using the inductive definition of A^* or by using the fact that A^* may be computed by WFK - algorithm. We choose the latter. It is enough to prove, with the notations used in Section 4 that for $k \in [0:n]$, $a^{(k)}$ is the sum of the labels of all non-empty elementary paths from i to j the intermediate vertices of which are in [1:k]. The assertion is proved by induction on k by simple algebraic manipulations.

Theorem 4: If A is a n×n matrix over a simple semi-ring

$$A^* = I_n + A + A^2 + \ldots + A^{n-1}$$

Proof: An elementary path has length less or equal to n-1 and the labels of all elementary paths of length ℓ are terms in some element of A^ℓ.

Conversely a term in an element of A^ℓ ($\ell \leqslant n$) which is the label of a non-elementary path is absorbed by a term of A^k for $k<\ell$ which is the label of a shorter elementary path.

Corollary: If B is the matrix obtained from A by interchanging rows i and j and columns i and j then B* is obtained from A* by the same exchanges.

This is not true for a general closed semi-ring but Conway has shown in [3] that it holds if the three following identities hold:

$$a.0 = 0.a = 0, \quad (a + b)^* = a^*(ba)^*, \quad (ab)^* = 1 + a(ba)^*b.$$

This implies that if $A = \begin{bmatrix} B & C \\ D & E \end{bmatrix}$ then

$$A^* = \begin{bmatrix} \Delta^* & \Delta^*CE^* \\ E^*D\Delta^* & E^* + E^*D\Delta^*CE^* \end{bmatrix} \quad \text{for } \Delta = B + CE^*D.$$

7. Dijkstra semi-rings and Dijkstra's algorithm.

Definition: A Dijkstra semi-ring is a simple semi-ring in which

i) $a + b = \begin{cases} a \\ b \end{cases}$ or

Note: It is easy to see that a Dijkstra semi-ring is totally ordered by the relation: $a \geqslant b$ iff $a + b = a$.

The addition is then a maximum operation in the ordered set:

a + b = the maximum of a and b.

The closure of a matrix over a Dijkstra semi-ring can be computed row by row by the following algorithm.

Dijkstra's algorithm [4]:

Input: $A = [a_{ij}]i,j\epsilon[1:n]$ a_{ij} elments of a Dijkstra semi-ring , or $\epsilon[1:n]$

```
     begin
1.       T ← {or} ;
2.   b [or] ← 1;
3.   for each   i∈[1:n] - {or} do  b[i] ← A[or, i];
4.   for each k∈[2:n] do
5.   find a j∈[1:n] - T such that b_j =      Σ          b_ℓ;
                                          ℓ∈[1:n]-T
6.       T ← T∪{j};
7.   for each   i∈[1:n] - T do  b[i] ← b[i] + b[j].A[j,i];
     end
         Output:  b[i]  ,   i∈[1:n]
```

Claim: the output $B = [b[i]]_{i\in[1:n]}$ is the orth row of A^*.

Notice that statement 5 has a clear meaning in a Dijkstra semi-ring because of property i).

Proof of correctness:

By the corollary to Theorem 4 we may supoose that or = 1 and that j in statement 5 is equal to k. Then the algorithm computes a sequence of n rows: $b^{(1)} \ldots b^{(n)}$, the last one being the output, such that:

$$b^{(1)} = \begin{bmatrix} 1 & A_{1[2:n]} \end{bmatrix}$$

$$b^{(k+1)} = b^{(k)} + b^{(k)}_{k+1} \begin{bmatrix} 0_{k+1} & A_{k+1[k+2:n]} \end{bmatrix} \qquad \text{for } 1 \leqslant k \leqslant n-1$$

where 0_k is a row of k zeroes and $b^{(k)}_{k+1}$ is such that

$$b^{(k)}_{k+1} = \sum_{\ell=k+1}^{n} b^{(k)}_{\ell}$$

The correctness of the algorithm follows from

Theorem 5: For any k $1 \leqslant k \leqslant n$

$$b^{(k)} = (A^*)_{1[1:k]} \begin{bmatrix} I_k & A_{[1:k][k+1:n]} \end{bmatrix}$$

where I_k is the identity matrix of size k.

$$\left(\text{equivalently: } b^{(k)} = \begin{bmatrix} (A^*)_{1[1:k]} & b^{(k)}_{[1:k]} A_{[1:k][k+1:n]} \end{bmatrix} \right)$$

Proof: By induction on k.

For $k = 1$, $b^{(1)} = 1 \begin{bmatrix} 1 & A_{1[2:n]} \end{bmatrix} = \begin{bmatrix} 1 & A_{1[2:n]} \end{bmatrix}$

because $(A^*)_{11} = 1$ for any A.

For $1 < k \leqslant n$: $b_k^{(k-1)} = \sum_{\ell=k}^{n} b_\ell^{(k-1)}$ and

$$b^{(k)} = b^{(k-1)} + b_k^{(k-1)} \begin{bmatrix} 0_k & A_{k[k+1:n]} \end{bmatrix}$$

$$= \begin{bmatrix} b^{(k-1)}_{[1:k]} & b^{(k-1)}_{[k+1:n]} + b_k^{(k-1)} A_{k[k+1:n]} \end{bmatrix}.$$

By the induction hypothesis:

$$b^{(k-1)}_{[k+1:n]} = b^{(k-1}_{[1:k-1]} A_{[1:k-1][k+1:n]}$$ and

$$b^{(k)} = \begin{bmatrix} b^{(k-1)}_{[1:k]} & b^{(k-1)}_{[1:k]} A_{[1:k][k+1:n]} \end{bmatrix}$$

It is left to prove that:

$$b^{(k-1)}_{[1:k]} = (A^*)_{1[1:k]}$$

By the induction hypothesis: $b^{(k-1)}_{[1:k-1]} = (A^*)_{1[1:k-1]}$

and

$$b_k^{(k-1)} = \sum_{\ell=k}^{n} b_\ell^{(k-1)}$$

$$= b_k^{(k-1)} + \sum_{\ell=k+1}^{n} b_\ell^{(k-1)} (1+B_\ell) \quad \text{for any column B}$$

$$= \sum_{\ell=k}^{n} b_\ell^{(k-1)} + b^{(k-1)}_{[k+1:n]} B_{[k+1:n]}$$

$$= b_k^{(k-1)} + b^{(k-1)}_{[k+1:n]} B_{[k+1:n]}$$

We may choose $B = \left(A_{[k:n][k:n]} \right)^*_{[2:n]1}$

then

$$b_k^{(k-1)} = b_{[k:n]}^{(k-1)} \left(A_{[k:n][k:n]} \right)^*_{.1}$$ because in a simple

semi-ring the diagonal of a closure matrix contains only ones.

By the corollary to Theorem 4 it is clear that:

$$(A^*)_{1k} = (A^*)_{1[1:k.1]} \; A_{[1:k-1][k:n]} \left(A_{[k:n][k:n]} \right)^*_{.1}$$

$$= b_k^{(k-1)}$$ by the induction hypothesis.

Q.E.D.

Notice that the hypothesis i) is not used at all in the proof of correctness it only guarantees that statement 5 of the algorithm is meaningful.

Acknowledgements

This work originated in discussions with Peter Wegner and during the first part of its elaboration I had a fruitful exchange with C. Elgot; I thank them for their encouragement.
Finally I am most grateful to Michael Paterson for pointing me earlier works on the subject, his suggestions and constant interest; if some measure of clarity has been achieved it is only thanks to him.

Appendix 1

Let us compute the closure of the square matrix

$$P = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \quad \text{where } A, E \text{ and } I \text{ are square}$$

using two different decompositions.

<u>Decomposition 1</u>:

$$P = \left[\begin{array}{cc|c} A & B & C \\ D & E & F \\ \hline G & H & I \end{array}\right]$$

$$\Delta_1 = I + [G\ H] \begin{bmatrix} A & B \\ D & E \end{bmatrix}^* \begin{bmatrix} C \\ F \end{bmatrix}$$

$$\begin{bmatrix} A & B \\ D & E \end{bmatrix}^* = \begin{bmatrix} A^* + A^*B\delta_1^*DA^* & A^*B\delta_1^* \\ \delta_1^*DA^* & \delta_1^* \end{bmatrix} \quad \text{for } \delta_1 = E + DA^*B$$

and

$$\Delta_1 = I + G[A^* + A^*B\delta_1^*DA^*]C + H\delta_1^*DA^*C + GA^*B\delta_1^*F + H\delta_1^*F .$$

$$= I + GA^*C + (H + GA^*B)\delta_1^*(F + DA^*C)$$

$$P^* = \begin{bmatrix} A' & B' & C' \\ D' & E' & F' \\ G' & H' & I' \end{bmatrix}$$

$$I' = \Delta_1^*$$

$$[G'\ H'] = \Delta_1^*[G\ H] \begin{bmatrix} A & B \\ D & E \end{bmatrix}^* = \Big[\Delta_1^*G(A^* + A^*B\delta_1^*DA^*) + \Delta_1^*H\delta_1^*DA^* ,$$

$$\Delta_1^*GA^*B\delta_1^* + \Delta_1^*H\delta_1^*\Big]$$

$$\begin{bmatrix} C' \\ F' \end{bmatrix} = \begin{bmatrix} A & B \\ D & E \end{bmatrix}^* \begin{bmatrix} C \\ F \end{bmatrix} \Delta_1^* = \begin{bmatrix} (A^* + A^*B\delta_1^*DA^*)C\Delta_1^* + A^*B\delta_1^*F\Delta_1^* \\ \delta_1^*DA^*C\Delta_1^* + \delta_1^*F\Delta_1^* \end{bmatrix}$$

$$\begin{bmatrix} A' & B' \\ D' & E' \end{bmatrix} = \begin{bmatrix} A & B \\ D & E \end{bmatrix}^* + \begin{bmatrix} A & B \\ D & E \end{bmatrix}^* \begin{bmatrix} C \\ F \end{bmatrix} \Delta_1^* \ [G\ H] \begin{bmatrix} A & B \\ D & E \end{bmatrix}^*$$

$$A' = A^* + A^*B\delta_1^*DA^* + [(A^* + A^*B\delta_1^*DA^*)C + A^*B\delta_1^*F]\ \Delta_1^*\ [G(A^* + A^*B\delta_1^*DA^*) + H\delta_1^*DA^*]$$

$$B' = A^*B\delta_1^* + [(A^* + A^*B\delta_1^*DA^*)C + A^*B\delta_1^*F]\ \Delta_1^*\ [GA^*B\delta_1^* + H\delta_1^*]$$

$$D' = \delta_1^*DA^* + [\delta_1^*DA^*C + \delta_1^*F]\ \Delta_1^*\ [G(A^* + A^*B\delta_1^*DA^*) + H\delta_1^*DA^*]$$

$$E' = \delta_1^* + [\delta_1^*DA^*C + \delta^*F]\ \Delta_1^*\ [GA^*B\delta_1^* + H\delta_1^*]$$

$$= \delta_1^* + \delta_1^*\ [F + DA^*C]\ \Delta_1^*\ [H + GA^*B]\delta_1^*$$

<u>Decomposition 2</u>:

$$P = \left[\begin{array}{c|cc} A & B & C \\ \hline D & E & F \\ G & H & I \end{array}\right]$$

$$\Delta_2 = \begin{bmatrix} E & F \\ H & I \end{bmatrix} + \begin{bmatrix} D \\ G \end{bmatrix} A^* \quad [B\ C]$$

$$= \begin{bmatrix} E + DA^*B & F + DA^*C \\ H + GA^*B & I + GA^*C \end{bmatrix} = \begin{bmatrix} \delta_1 & F + DA^*C \\ H + GA^*B & I + GA^*C \end{bmatrix}$$

$$\delta_2 = I + GA^*C + (H + GA^*B)\delta_1^*(F + DA^*C) = \Delta_1$$

$$\Delta_2^* = \begin{bmatrix} \delta_1^* + \delta_1^*(F + DA^*C)\ \Delta_1^*\ (H + GA^*B)\delta_1^* & \delta_1^*(F + DA^*C)\Delta_1^* \\ \Delta_1^*(H + GA^*B)\delta_1^* & \Delta_1^* \end{bmatrix}$$

$$I' = \Delta_1^*$$

$$H' = \Delta_1^*(H + GA^*B)\delta_1^*$$

$$F' = \delta_1^*(F + DA^*C)\Delta_1^*$$

$$E' = \delta_1^* + \delta_1^*(F + DA^*C)\Delta_1^* (H + GA^*B)\delta_1^*$$

$$\begin{bmatrix} D' \\ G' \end{bmatrix} = \Delta_2^* \begin{bmatrix} D \\ G \end{bmatrix} A^*$$

$$D' = ([\delta_1^* + \delta_1^*(F + DA^*C)\ \Delta_1^*\ (H + GA^*B)\delta_1^*]\ D + \delta_1^*(F + DA^*C)\Delta_1^*G\ A^*$$

$$= \delta_1^*DA^* + \delta_1^*(F + DA^*C)\ \Delta_1^*\ (GA^* + (H + GA^*B)\delta_1^*DA^*)$$

$$G' = \Delta_1^*\ (H + GA^*B)\delta_1^*DA^* + \Delta_1^*GA^*$$

$$[B'\ C'] = A^*[B\ C]\Delta_2^*$$

$$B' = A^*B\left(\delta_1^* + \delta_1^*(F + DA^*C)\ \Delta_1^*\ (H + GA^*B)\delta_1^*\right) + A^*C\Delta_1^*(H + GA^*B)\delta_1^*$$

$$= A^*B\delta_1^* + \left[(A^* + A^*B\delta_1^*DA^*)C + A^*B\delta_1^*F\right]\Delta_1^*\ (H + GA^*B)\delta_1^*$$

$$C' = A^*B\delta_1^*(F + DA^*C)\Delta_1^* + A^*C\Delta_1^*$$

$$A' = A^* + A^*[B\ C]\ \Delta_2^* \begin{bmatrix} D \\ G \end{bmatrix} A^*$$

$$= A^* + A^* \Big[B\left(\delta_1^* + \delta_1^*(F + DA^*C)\ \Delta_1^*\ (H + GA^*B)\delta_1^*\right) D + C\Delta_1^*(H + GA^*B)\delta_1^*D$$

$$+ B\delta_1^*(F + DA^*C)\Delta_1^*G + C\Delta_1^*G\Big]\ A^*$$

$$= A^* + A^*B\delta_1^*DA^* + A^*\left[\left(B\delta_1^*(F + DA^*C) + C\right)\ \Delta_1^*\left((H + GA^*B)\delta_1^*D + G\right)\right]A^*$$

Comparing the expressions obtained for the elements of P* by both methods one sees that they are equivalent.

Appendix 2

Before we proceed to the proof of the fundamental property of simple semi-rings some terminology is required.

A n×n matrix can be viewed as a labelled complete directed graph on n vertices. a_{ij} is the label of edge (i,j).

To any path $(k_o, k_1, k_2, \ldots, k_m)$ from k_o to k_m in the graph can be associated a unique label: $a_{k_o k_1} a_{k_1 k_2} \cdots a_{k_{m-2} k_{m-1}} a_{k_{m-1} k_m}$

The label 1 is associated with the empty path from k_o to k_o, for all k_o.

The fundamental property of simple semi-rings says that the (i,j)-th element of the closure of a matrix A is the sum of the labels of all <u>elementary</u> paths from i to j.

<u>Proof:</u> We could use the inductive definition of A* but we prefer to use WFK-algorithm.

$R = A^* = I_n + A^{(n)}$ and it is left to show

that $$a_{ij}^{(n)} = \sum_{\substack{m \\ k_1,\ldots,k_m \in [1:n] \\ k_1,\ldots k_m \text{ all distinct} \\ \text{and different from } i \text{ and } j.}} a_{ik_1} \cdots a_{k_m j}$$ = the sum of the labels of all non-empty elementary paths from i to j.

We shall prove that for any $\ell \in [0:n]$

$$a_{ij}^{(\ell)} = \sum_{\substack{m \\ k_1,\ldots k_m \in [1:\ell] \\ k_1,\ldots k_m \text{ all distinct} \\ \text{and different from } i \text{ and } j.}} a_{ik_1} \cdots a_{k_m j} \underset{def}{=} \text{sigma}(\ell,i,j)$$

= the sum of the labels of all non-empty elementary paths from i to j the intermediate vertices of which are in $[1:\ell]$.

For $\ell = 0$ $a_{ij}^{(o)} = a_{ij} =$ sigma (0,i,j)

For $\ell = h+1$. By WFK-algorithm: $a_{ij}^{(\ell)} = a_{ij}^{(h)} + a_{i\ell}^{(h)} \cdot a_{\ell j}^{(h)}$

because $\left(a_{\ell\ell}^{(h)}\right) = 1.$

By the induction hypothesis:

$$a_{ij}^{(\ell)} = \text{sigma } (h,i,j) + \text{sigma } (h,i,\ell).\text{sigma } (h, \ell,j)$$

Let us consider sigma (ℓ,i,j): its terms are labels of non-empty elementary paths from i to j the intermediate vertices of which are in [1:h+1]; of those the labels of the paths of which $\ell = h+1$ is not an intermediate vertex are terms in sigma (h,i,j) and the labels of the paths of which ℓ is an intermediate vertex are terms in the product sigma (h,i,ℓ).sigma (h,ℓ,j) because ℓ is visited only once in such an elementary path.

Conversely all the terms of sigma (h,i,j) appear in sigma (h+1,i,j); suppose $a_{ik_1} \dots a_{k_m \ell} \cdot a_{\ell h_1} \dots a_{h_m, j}$ is a term of sigma (h,i,ℓ).sigma (h,ℓ,j).

If $(i,k_1,\dots,k_m,\ell,h_1,\dots,h_m,j)$ is an elementary path it appears in sigma (ℓ,i,j) if it is not elementary there is a vertex v which is visited twice (and at least once as an intermediate node):

$$v = k_s = h_t \quad , \quad v = i = h_t \quad \text{or } v = k_s = j.$$

Let s be the smallest integer such that k_s is visited twice and let t be the largest integer such that $h_t = k_j$. Then $(i,k_1,\dots,k_s,h_{t+1},\dots,j)$ is a non-empty elementary path from i to j which does not go through ℓ and its label appears in sigma (h,i,j).

By the identities:

$$a_{ik_1}\dots a_{k_{s-1}k_s}\, a_{h_t h_{t+1}}\dots a_{h_m,j} + a_{ik_1}\dots a_{k_m \ell}\, a_{\ell h_1}\dots a_{h_m,j}$$

$$= a_{ik_1}\dots a_{k_{s-1}k_s} \left[1 + a_{k_s k_{s+1}}\dots a_{h_{t-1}h_t} \right] a_{h_t h_{t+1}}\dots a_{h_m,j}$$

$$= a_{ik_1}\dots a_{k_{s-1}k_s}\, a_{h_t h_{t+1}}\dots a_{h_m,j}$$

All the terms appearing in sigma (h,i,ℓ). sigma (h,ℓ,j) which are not in sigma (ℓ,i,j) are absorbed by terms of sigma (h,i,j) and:

$$\text{sigma}\ (\ell,i,j) = \text{sigma}\ (h,i,j) + \text{sigma}\ (h,i,\ell).\text{sigma}\ (h,\ell,j)$$

and $a_{ij}^{(\ell)} = \text{sigma}\ (\ell,i,j)$ Q.E.D.

Bibliography

[1] Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, The design and analysis of computer algorithms (Addison-Wesley, Reading, Mass. 1974) 195-201.

[2] R.C. Backhouse and B.A. Carré, Regular agebra applied to path-finding problems, J. Inst. Maths. Applics., 15 (1975), 161-186.

[3] J.H. Conway, Regular algebra and finite machines (Chapman and Hall, London 1971) 109-111.

[4] E.W. Dijkstra, A note on two problems in connection with graphs, Numerische Mathematik 1 (1959) 269-271.

[5] R.W. Floyd, Algorithm 97: shortest path, Comm. ACM5:6 (1962) 345.

[6] M. Gondran, Algèbre linéaire et Cheminement dans un graphe, R.A.I.R.O. 9^{e} annee (janvier 1975)V-1 p.77-99

[7] John B. Kam and Jeffrey D. Ullman, Global Data Flow Analysis and Iterative Algorithms JACM Vol. 23. No.1. January 76.

[8] S.C. Kleene, Representation of events in nerve nets and finite automata, in C.E. Shannon and J. McCarthy (eds.) Automata Studies, 3-42 (1956) Princeton University Press. Princeton N.J.

[9] Daniel J. Lehmann, Algebraic structures for transitive closure, (Theory of Computation Report No.10, Department of Computer Science, The University of Warwick, February 1976)

[10] B. Roy, Transitivité et connexité, C.R. Acad. Sciences de Paris 249 (1959) p.216.

[11] R.E. Tarjan, Solving path problems on directed graphs, (Technical Report, Stanford Computer Science Department, October 1975)

[12] R.E. Tarjan, Graph theory and Gaussian elimination in J. Bunch and D. Rose, eds., Sparse matrix computations (Academic Press, New York, to appear)

[13] S. Warshall, A theorem on Boolean matrices, J. ACM 9:1 (1962) -11-12.

[14] M. Yoeli, A note on a generalisation of boolean matrix theory, American Mathematical Monthly, Vol. 68 1961 552-557.