Zeitschrift: L'Enseignement Mathématique
Herausgeber: Commission Internationale de l'Enseignement Mathématique
Band: 26 (1980)
Heft: 1-2: L'ENSEIGNEMENT MATHÉMATIQUE

Artikel: THE FAST SKEW-CLOSURE ALGORITHM
Autor: Fischer, M. J. / Paterson, M. S.
Kapitel: 2. The Boolean algorithm
DOI: https://doi.org/10.5169/seals-51079

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:

$$\text{for } i := 1 \text{ step } 1 \text{ until } n - 1 \text{ do } \psi_i$$

where ψ_i is

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

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

$$\text{for } i := n \text{ step } -1 \text{ until } 2 \text{ do}$$

and

$$k := i - 1 \text{ step } -1 \text{ until } (k = 1 \ldots$$

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^{Ω}, of A is given by

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

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

THEOREM 1.

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

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

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

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

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

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

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