American Research Journal of Computer Science and Information Technology

Volume 5, Issue 1, 1-4 Pages
Research Article | Open Access
ISSN (Online)- 2572-2921
DOI :

AMERICAN RESEARCH JOURNALS
An Academic Publishing House

New Transitive Closure Criterion

Marius Orlowski

Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, Virginia 24061, USA.

ABSTRACT

Novel recursive matrix algorithm for the transitive closure is proposed that is computationally more efficient than the closure theorem $O(n^4)$ and runs only slightly longer in time $O(n^3 ln_2(n))$ than the Warshall algorithm $O(n^3)$. However, the proposed algorithm can be combined with advanced, extant algorithms by Chan, Zwick, and Margalit to further improve its bit operation performance. From the recursive matrix algorithm a simple criterium can be derived to ascertain whether some given adjacency matrix of a graph with n nodes already represents a transitive closure. If the criterium is met, the calculation of the transitive closure by any other algorithm can, obviously, be eschewed.

INTRODUCTION

Many problems in science and engineering may be cast in terms of directed or undirected graphs. Designing algorithms for solving graph problems is therefore both of practical importance and of theoretical interest. Finding the transitive closure of a binary relation over a finite set or of a corresponding graph is widely used technique in many areas of computing. The well-known Roy-Floyd-Warshall algorithm [1-6] solves the problem on a sequential computer architecture in $O(n^3)$ steps, where n is the dimension of the underlying set of vertices.

Finding the transitive closure of a directed graph is an important problem in many computational tasks. It is required, for instance, in the reachability analysis of transition networks representing distributed and parallel systems and in the construction of parsing automata in compiler construction. Recently, efficient transitive closure computation has been recognized as a significant subproblem in evaluating recursive database queries, since almost all practical recursive queries are transitive [7-10].

Transitive closure operation is also applied to compiler construction for parsing automata [11,12]. A major issue in transitive closure algorithms is the avoidance of time-consuming redundant operations. Transitive closure is also closely associated with all-pairs shortest paths problems where shortest paths between arbitrary vertex pairs need to be determined [10, 13].

The transitive closure of a directed graph is defined as a graph in which an edge exists between two nodes A and B, if and only if there is a path from A to B in the original path. Given a directed graph, G(V,E) with V denoting the set of vertices and E denoting the set of edges in G, the transitive closure G* of G is defined as G*(V,E*) where E* contains edges (A,B) if and only if there exists a directed path from vertex A to vertex B in the original G. Finding the transitive closure of a binary relation over a finite set or of a corresponding graph is widely used technique in many areas of computing. The problem can be solved by the Floyd-Warshall algorithm [1-6] or by repeated breadth-first search [8,14] or depth-first search [15] starting from each node of the graph.

Transitive closure is also a fundamental operation in solving reachability problems for database querying and in reachability analysis of transition networks in distributed systems.

TRANSITIVE CLOSURE ALGORITHMS

The "squaring" algorithm for transitive closure states that given an adjacency matrix A of a graph G with n vertices the transitive closure matrix W* is given by

$$W^*=A+A^2+ \ldots +A^n \qquad \text{eq.(1)}$$

using Boolean operations OR (+) and AND (·) and where $A \cdot A^j = A^{(j+1)}$. The computation time for eq.(1) is $O(n^4)$ bit operations. The well-known Roy-Warshall-Floyd [1-3] algorithm reduced the computation time from $O(n^4)$ to $O(n^3)$. Given the adjacency *n x n* matrix A with matrix elements a(i,j) of a graph G, the Warshall algorithm computes the adjacency matrix W* of the transitive closure of G by the following algorithm:

$$a^{(p+1)}(i,j):=a^p(i,j)+a^p(i,k) \cdot a^p(k,j) \quad 1 \le i,j,k \le n \qquad \text{eq.(2)}$$

For convenience, the sign · will be suppressed in the following. The algorithm constructs a sequence of adjacency matrices $W_o, \ldots W_n$, where $W_o = A$ and each W_k represents all paths of G containing no intermediate vertices higher than k. For k>0, W_k is obtained from W_{k-1} by looking up at all pairs of vertices (i,j) if there is a path from *i* to *j* representedin W_{k-1}, or if there are paths represented in W_{k-1} going from i to k and from *k* to *j*. Since no path can contain a vertex higher than n $W_n = W^*$. Many improvements of the basic Warshall algorithms have been proposed since then. The transitive closure problem is very well known [8,9] since Roy, Dijstra, Warshall, and Warren's early work [1-4,16]. Recently, the complexity of the problem of the transitive closure has been rediscovered and new advanced algorithms have been developed in the pioneering work by Zwick [17-19], Margalit [20,21] and Chan [23-26] and others [27].

PROPOSED ALGORITHM FOR TRANSITIVE CLOSURE

If A is the adjacency *n x n* matrix of the direct graph G then the novel algorithm is based on the following recursive relation

$$W_o = A$$

$$W_{(k+1)} = W_k + W_k \cdot W_k \quad k=0, \lceil \ln_2 n \rceil \quad eq.(3)$$

where ⌈x⌉ denotes the rounding of x∊R to the next larger integer. The proof that the recursive generation of matrices W_k leads to the transitive closure is readily demonstrated. Any W_k, obtained by k iterations form eq. (3) for *k* larger than 1 can be shown to be represented by the sum:

$$W_k = W_o + \sum_{k=1}^{m_2} W_o^2 + \cdots + \sum_{k=1}^{m_{k^2-1}} W_o^{k^2-1} + W_o^{k^2} \qquad eq.(4)$$

Where $m_j \in N$ are integer numbers arising from the recursive relation (3). However, because the logic OR operation on any binary matrix A is idempotent, i.e. A=A+A, then

$$\sum_{i=1}^{N} W_o^k = W_o^k \qquad eq.(5)$$

for any k,N≥ 2. Hence, the eq.(4) can be written in a more compact form as:

$$W_k = \sum_{l=1}^{k^2} W_o^l \qquad eq.(6)$$

If $n \le k^2$ then according to the transitive closure theorem the transitive closure has been reached. Thus, it is sufficient to do the iterations up to $k = \lceil \ln_2 n \rceil$. Hence, the matrix for transitive closure W* is given by

$$W^* = \sum_{l=1}^{\lceil \ln_2 n \rceil} W_o^l \qquad eq.(7)$$

For all n that can be expressed as $n=2^k$ with k∈N the rounding expression can be replaced by the argument itself. It is interesting to note that the factor in the runtime of $\ln_2 n$ has also been found in the context of related studies [27,28], albeit under different circumstances. Thus, the proposed algorithm is equivalent to the squaring algorithm given in eq.(1) but has the advantage of avoiding calculation of all power matrices A^k for k=1,n. Moreover, because of the recursive relation, from the preceding analysis the following criterium for the status of the transitive closure can be derived. For a given matrix A one can ascertain using the operation

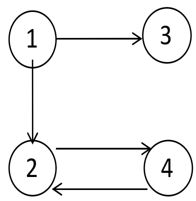
$$A = A + A \cdot A \qquad eq.(8a)$$

whether A represents indeed the transitive closure. If eq.(8a) holds the A must represent the transitive closure, i.e. A=W*. If A does not represent transitive closure then one obtains the inequality

$$A \neq A + A \cdot A \qquad eq.(8b)$$

Then, obviously, A≠W* and some more iterations of the recursive formula have to be computed to reach the transitive closure. This may be of great use in case of sparse matrices or in cases where reasonable guess may be warranted or sufficient extant information available that the transitive closure may have already been found or is being close of being attained. Thus, in many practical cases, where the reachability matrix is at least partially guaranteed, the above approach may effectively reduce the computational burden to a runtime of $O(n^2 \ln_2 n)$. Consequently, it is not necessary to apply the brute force method of the Warshall of $O(n^3)$ or more complex algorithms of $O(n^s)$ with s<3.

As an illustration consider the graph in Fig.1 with the associated adjacency Boolean matrix A (Fig.1).



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

Fig.1. Graph of 4 vertices with directed edges and the corresponding adjacency matrix.

In the traditional closure theorem one would calculate in addition to A calculate one would have to calculate the power matrices A^2, A^3, A^4 given as

$$A^2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}, A^3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}, A^4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix},$$

If one adds up $A+A^2+A^3+A^4$ one gets the transitive closure W*.

$$W^* = A^1 + A^2 + A^3 + A^4 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

For the Warshall algorithm the respective matrices to be calculated are A_1, A_2, A_3, and A_4 which are given by

$$\begin{pmatrix}0&0&1&0\\1&0&1&0\\0&0&0&0\\0&1&0&0\end{pmatrix}\begin{pmatrix}0&0&1&0\\1&0&1&1\\0&0&0&0\\1&1&1&1\end{pmatrix}\begin{pmatrix}0&0&1&0\\1&0&1&1\\0&0&0&0\\1&1&1&1\end{pmatrix}\begin{pmatrix}0&0&1&0\\1&1&1&1\\0&0&0&0\\0&1&1&1\end{pmatrix}$$

respectively, and the last matric A_4=W* is the transitive closure. Using algorithm proposed here and given in eq.(3) one obtains W_1 and W_2 as

$$W_1 = \begin{pmatrix}0&0&1&0\\1&1&1&1\\0&0&0&0\\1&0&0&1\end{pmatrix}, W_2 = \begin{pmatrix}0&0&1&0\\1&1&1&1\\0&0&0&0\\1&1&1&1\end{pmatrix}$$

where the second matrix W_2 is already the transitive closure, i.e. W_2=W*.

Let's now assume that an adjacency matrix of a directed graph with 5 vertices is given as

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

To make sure to obtain transitive closure for the underlying graph, we would have to apply the Warshall algorithm or a one of the more recent algorithms to calculate the transitive closure, W*. However, we may surmise by inspection (easy in this case) or have sufficient information that matrix A is already the transitive closure. One could confirm this supposition by performing the relatively simple operation given in eq. (8). Indeed, in this case, we'll find that A=A+AA and the burden of a calculation by Warshall or any other direct or indirect algorithm can be safely avoided.

CONCLUSION

A new recursive matrix algorithm has been proposed that is equivalent to the transitive closure theorem and to Warshall algorithm and its successors. Its run time is $O(n^2 \ln_2 n)$ which is slightly longer than Warshall-Floyd algorithm and hence not particularly attractive, considering the recently proposed much more involved logarithms, some of which have a runtime as low as of $\ln n^{2.49}$ [28,29] as described above. However, the proposed matrix algorithm can take advantage of the recent more advanced algorithms and spawn potential new synergies. The main benefit of the algorithm, in our view, appears to be the criterium given in eq. (8) that allows to check whether a given adjacency matrix may already represent the transitive closure or may be very close to the transitive closure. If the given adjacency matrix does not fulfill the transitive closure criterium, i.e., $A \neq A+A \cdot A$, then the objective of the future work is to find out, how close is A is, in terms of recursive loops in eq. (3), to the final solution W*. It may be surmised that it should be possible to derive such a proximity criterium to reach the final solution from a few iterations using the relation given in eq. (3).

REFERENCES

1. S. Warshall, "A Theorem on Boolean Matrices," J. Assoc. Comput. Mach. 9, pp. 11-12, 1962
2. B. Roy, "Trasitivite et Connexite," C.R. Acad. Sci. Paris Der. A-B 249, pp.216-218, 1959
3. W. R. Floyd, "Algorithm 97: Shortest path," Comm. ACM, 5,pp. 344-345, 1962
4. E. W. Dijkstra, "A Note on two Problemsinconnexion with Graphs," Num. Math., 1,pp..269-271, 1959
5. C. P. Schnorr, An Algorithm for Transitive Closure with Linear Expected Time,' SIAM J. Computing, 7(2), pp. 127-133, 1978
6. H. Poor, "A Hypertext History of Multiuser Dimensions," *MUD History,* http://www.ccs.neu.edu/home/pb/mud-history.html. 1986.
7. T. Nadueau, K. Gray, *Software Defined Networking*, O'Reilly, USA, 2013
8. T. Comen, C. Leiserson, R. Rivest, S. Stein,K. Elissa, Introduction to Algorithms, MIT Press, USA, 2009
9. B. Chazelle, "Cuttings" in Handbook of *Data Structures and Applications*, CRC Press, Boca Raton, pp. 25.1-25.10, 2005
10. A. Aho, J. E. Hopcroft, J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 2008
11. A. V. Aho, R. Sethi, J. D. Ulmann, "Compilers, Principles, Techniques", Addison Wesley, 7(8), p. 9, 1986
12. S. Sippu, E. Soisalon-Soinin, Langages and Parsing, Springer, Berlin 1988
13. S. Y. Kung, S.C. Lo, P.S. Lewis, 'Optimal Systolic Design for the Transitive Closure and the Shortest Path Problems,' IEEE Trans. Comput. C36(5), pp. 603-614 , 1987
14. *S. Skiena,* "Sorting and Searching". *The Algorithm Design Manual,* Springer. p. 480, 2008
15. E. Shimon, *Graph Algorithms,* (2nd ed.), Cambridge University Press, pp. 46–48, 2011
16. H. S. Warren, "A Modification of Warshall's Algorithm for the Transitive Closure of Binary Relations," Comm. ACM 18(4), pp. 218-220, 1975
17. U. Zwick, All pairs lightest shortest paths, in Proceedings of the 31st ACM Symposium Theory of Computing, Atlanta, pp. 61–69, 1999
18. U. Zwick, "All-pairs shortest paths using bridging sets and rectangular matrix multiplication," J. ACM, 49, pp. 289–317, 2002
19. U. Zwick, "A slightly improved sub-cubic algorithm for

all pairs shortest paths problem with real edge lengths," Algorithmica, 46 pp. 181–192, 2006

20. Z. Galil, O. Margalit, "All Pairs Shortest Paths for Graphs with Small Integer Length Edges", J. Computer and Syst. Sci., 54 pp.243-254, 1997
21. Zvi Galil, Oded Margalit:"Witnesses for Boolean Matrix Multiplication and for Transitive Closure," J. Complex., 9(2), pp.201-221, 1993
22. T. M. Chan, "Dynamic subgraph connectivity with geometric applications," SIAM J. Comput., 36, pp. 681-694, 2006
23. T. M. Chan, "All-pairs shortest paths with real weights in $O(n^3/\log n)$ time", Algorithmica, 50, pp. 236-243, 2008
24. T. M. Chan and A. Efrat, "Fly cheaply: On the minimum fuel consumption problem, "J. Algorithms, 41, pp. 330–337, 2002
25. T. M. Chan, "More Algorithms for all pairs Shortest Path in Weighted Graphs," SIAM J. Comput., 39(5), pp. 2075-2089, 2010
26. P. E. O'Neil, E. J. O'Neil, "A Fast Expected Time Algorithm for Boolean Matrix," Information and Control, 22, pp. 132-138, 1973
27. M. Benedikt, P. Senellart, P. "Databases", in Blum, Edward K.; Aho, Alfred V. (eds.). *Computer Science. The Hardware, Software*, pp. 169–229, 2011
28. S. Sridhar, *Design and Analysis of Algorithms*, Oxford University Press, 2014
29. Z. Zuo, X. Liu, Q. Huang, Y. Liao, Y. Wang, and C. Wang, "Derivation and Formal Proof of Floyd-Warshall Algorithm," *2021 5th International Conference on Communication and Information Systems (ICCIS)*, 2021, pp. 202-207, 2021

Citation: Marius Orlowski, "New Transitive Closure Criterion", American Research Journal of Computer Science and Information Technology, Vol 5, no. 1, 2022, pp. 1-4.