SIAM J. COMPUT.
Vol. 1, No. 2, June 1972

# THE TRANSITIVE REDUCTION OF A DIRECTED GRAPH*

A. V. AHO,† M. R. GAREY† AND J. D. ULLMAN‡

**Abstract.** We consider economical representations for the path information in a directed graph. A directed graph $G'$ is said to be a transitive reduction of the directed graph $G$ provided that (i) $G'$ has a directed path from vertex $u$ to vertex $v$ if and only if $G$ has a directed path from vertex $u$ to vertex $v$, and (ii) there is no graph with fewer arcs than $G'$ satisfying condition (i). Though directed graphs with cycles may have more than one such representation, we select a natural canonical representative as the transitive reduction for such graphs. It is shown that the time complexity of the best algorithm for finding the transitive reduction of a graph is the same as the time to compute the transitive closure of a graph or to perform Boolean matrix multiplication.

**Key words and phrases.** Directed graph, binary relation, minimal representation, transitive reduction, algorithm, transitive closure, matrix multiplication, computational complexity.

**1. Introduction.** Given a directed graph $G$, one is often interested in knowing whether there is a path from one vertex to another in that graph. In many cases it is possible to represent this information by another directed graph that has fewer arcs than the given graph. Informally, we say that a graph $G'$ is a transitive reduction of the directed graph $G$ whenever the following two conditions are satisfied:

(i) there is a directed path from vertex $u$ to vertex $v$ in $G'$ if and only if there is a directed path from $u$ to $v$ in $G$, and

(ii) there is no graph with fewer arcs than $G'$ satisfying condition (i).

Such minimal representations for graphs are of particular interest for efficiently executing certain computer algorithms, such as the precedence constrained sequencing algorithms of [1] and [2], whose operation is partially determined by an input-specified transitive relation. In particular, these minimal representations may require less computer memory for storage and, depending upon the precise nature of the algorithm, may also lead to a reduced execution time.

In this paper, we mathematically characterize the transitive reduction and provide an efficient algorithm for computing the transitive reduction of any given directed graph. Furthermore, we show that the computational complexity of computing a transitive reduction is equivalent to the computational complexity of computing a transitive closure or performing a Boolean matrix multiplication.

In [3], the *minimum equivalent* of a directed graph $G$ is defined as a smallest subgraph $G'$ of $G$ such that there is a path from vertex $u$ to vertex $v$ in $G'$ whenever there is a path from $u$ to $v$ in $G$. Our notion of transitive reduction is similar, but with the important exception that we do not require a transitive reduction to be a subgraph of the original graph. The two notions give rise to the same reduced representation when the original graph is acyclic. However, the transitive reduction of a graph $G$ with cycles can be smaller and much easier to find than a minimal equivalent graph for $G$.

* Received by the editors August 9, 1971, and in revised form November 15, 1971.

† Bell Telephone Laboratories, Inc., Murray Hill, New Jersey 07974.

‡ Department of Electrical Engineering, Princeton University, Princeton, New Jersey 08540.

The work of this author was supported in part by the National Science Foundation under Grant GJ-1052.

**2. Definitions and basic results.** A *directed graph* $G$ on the set of vertices $V = \{v_1, v_2, \cdots, v_n\}$ is a subset of $V \times V$, the members of $G$ being called *arcs*. A *directed path* in $G$ from vertex $u$ to vertex $v$ is a sequence of distinct arcs $\alpha_1, \alpha_2, \cdots, \alpha_p, p \geqq 1$, such that there exists a corresponding sequence of vertices $u = v_0, v_1, v_2, \cdots, v_p = v$ satisfying $\alpha_k = (v_k, v_{k+1}) \in G$, for $0 \leqq k \leqq p - 1$. A *cycle* is a directed path beginning and ending at the same vertex which passes through at least one other vertex. A *simple* cycle is a cycle which passes through no vertex more than once. A *loop* is an arc of the form $(v, v)$. A graph will be called acyclic if and only if it contains no cycles. Notice that this differs slightly from conventional usage, since we do allow an acyclic graph to contain loops.

A graph $G$ is said to be *transitive* if, for every pair of vertices $u$ and $v$, not necessarily distinct, $(u, v) \in G$ whenever there is a directed path in $G$ from $u$ to $v$. The *transitive closure* $G^T$ of $G$ is the least subset of $V \times V$ which contains $G$ and is transitive.

THEOREM 1. *For any finite acyclic directed graph $G$, there is a unique graph $G^t$ with the property that $(G^t)^T = G^T$ and every proper subset $H$ of $G^t$ satisfies $H^T \neq G^T$. The graph $G^t$ is given by*

$$G^t = \bigcap_{G_i \in S(G)} G_i,$$

*where $S(G) = \{G_1 | G_1^T = G^T\}$.*

*Proof.* The proof of Theorem 1 follows from the following two lemmas. (Note that Lemma 1 is actually a straightforword consequence of Theorem 1 in [3].)

LEMMA 1. *Let $G_1$ and $G_2$ be any two finite acyclic directed graphs (on the same vertex set) satisfying $G_1^T = G_2^T$. If there exists an arc $\alpha \in G_1$ such that $\alpha \notin G_2$, then $(G_1 - \{\alpha\})^T = G_1^T = G_2^T$.*

*Proof.* Let $\alpha = (u, v)$ be as described in the hypothesis of the lemma. Since $G_1^T = G_2^T$ and $\alpha \notin G_2$, $G_2$ must contain a path from $u$ to $v$ passing through some other vertex, say $w$. Then $G_1$ must contain a directed path from $u$ to $w$ and a directed path from $w$ to $v$. If the path from $u$ to $w$ in $G_1$ includes arc $\alpha$, then $G_1$ contains a path from $v$ to $w$. But, since $G_1$ also contains a path from $w$ to $v$, this contradicts $G_1$ being acyclic. If the path from $w$ to $v$ in $G_1$ includes arc $\alpha$, then $G_1$ contains a path from $w$ to $u$. But, since $G_1$ contains a path from $u$ to $w$, this also contradicts $G_1$ being acyclic. Thus, $G_1$ contains a directed path from $u$ to $w$ and from $w$ to $v$, which does not include arc $\alpha$. Therefore, $G_1 - \{\alpha\}$ contains a path from $u$ to $v$, so $(G_1 - \{\alpha\})^T = G_1^T = G_2^T$.

LEMMA 2. *Let $G$ be any finite acyclic directed graph. Then the set $S(G) = \{G_1 | G_1^T = G^T\}$ is closed under union and intersection.*

*Proof.* Let $G_1$ and $G_2$ be any two members of $S(G)$. Since $G_1^T = G_2^T = G^T$, $G_1 \cup G_2 \subseteq G^T$. Because $G^T$ is transitive, we then have $(G_1 \cup G_2)^T \subseteq G^T$. Furthermore, $G_1$ is contained in $G_1 \cup G_2$, so $G_1^T = G^T \subseteq (G_1 \cup G_2)^T$. Therefore, $(G_1 \cup G_2)^T = G^T$ and $(G_1 \cup G_2) \in S(G)$.

Now let $\{\alpha_1, \alpha_2, \cdots, \alpha_r\} = G_1 - (G_1 \cap G_2)$. By repeated application of Lemma 1, we have

$$\begin{aligned}
(G_1 - \{\alpha_1\})^T &= G_1^T, \\
(G_1 - \{\alpha_1\} - \{\alpha_2\})^T &= G_1^T, \\
&\vdots \\
(G_1 - \{\alpha_1\} - \{\alpha_2\} - \cdots - \{\alpha_r\})^T &= G_1^T.
\end{aligned}$$

But the last equation merely says that $(G_1 \cap G_2)^T = G_1^T = G^T$, so $(G_1 \cap G_2) \in S(G)$.

Since $S(G)$ is a finite set, Theorem 1 is obtained as a straightforward application of Lemma 2.

Theorem 1 shows that the intuitive definition of transitive reduction actually yields a unique graph for any finite acyclic directed graph. Furthermore, the transitive reduction of any such graph $G$ can be obtained by successively examining the arcs of $G$, in any order, and deleting those arcs which are "redundant," where an arc $\alpha = (u, v)$ is redundant if the graph contains a directed path from $u$ to $v$ which does not include $\alpha$.

We now extend this analysis to graphs which contain cycles. Consider the graph $G = \{(v_1, v_2), (v_2, v_3), (v_3, v_2), (v_2, v_1)\}$ of Fig. 1(a). If $H$ is any proper subset of $G$, then $H^T \neq G^T$. Thus, $G$ is its own minimum equivalent graph. However, the graph $G_1 = \{(v_1, v_2), (v_2, v_3), (v_3, v_1)\}$ of Fig. 1(b) contains only three arcs and has the same transitive closure as $G$, as does the graph $G_2 = \{(v_1, v_3),$

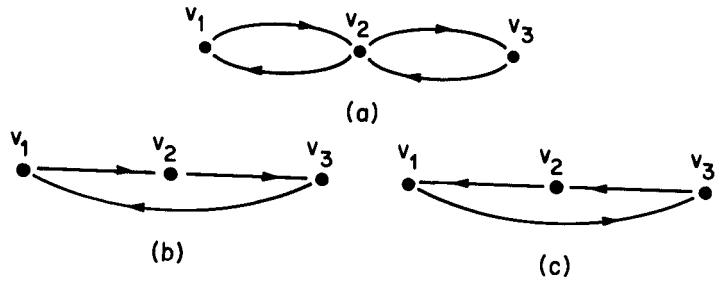


FIG. 1. *Graphs with cycles*

$(v_3, v_2), (v_2, v_1)\}$ of Fig. 1(c). No other graph with three or fewer arcs has the same transitive closure as $G$. Since $G_1 \neq G_2$, we see that for graphs with cycles there may not be a unique graph, with fewest arcs, having the same transitive closure as a given graph. Thus, Theorem 1 cannot be simply extended to encompass all finite directed graphs. This example also shows that the lemmas cannot be similarly extended. Furthermore, since neither $G_1$ nor $G_2$ is a subset of $G$, it may also be possible that no such minimal graph can be constructed by removing certain arcs from the given graph, as was the case for acyclic graphs. However, we shall show that all such minimal graphs, for any specific given graph, must have similar structure, and based on this result we choose a unique representative to be the unique transitive reduction.

Two vertices $u$ and $v$ of a directed graph $G$ will be called *equivalent* if either $u = v$ or $G$ contains a cycle which is incident with both $u$ and $v$. Given any finite directed graph $G$, we say that $G_1$ is the *equivalent acyclic graph* for $G$ when the vertices of $G_1$ are the vertex equivalence classes of $G$, denoted by $E_k$, and the arcs of $G_1$ satisfy: $(E_i, E_j) \in G_1$ if and only if there exists an arc $(u, v) \in G_1$ such that $u \in E_i$ and $v \in E_j$. If $G_1$ is the equivalent acyclic graph for $G$, and $G_2$ is a subset of $G_1$, then the graph $G_3$ is a *cyclic expansion* of $G_2$ if and only if:

(i) $G_3$ has the same vertices as $G$;

(ii) $G_2$ is the equivalent acyclic graph for $G_3$;

(iii) for each multimember vertex equivalence class $E_i$ of $G$, $G_3$ contains a simple cycle incident with all vertices in $E_i$ and $G_3$ contains no other arcs between members of $E_i$; and

(iv) for each arc $(E_i, E_j) \in G_2$, $E_i \neq E_j$, there is exactly one arc $(u, v) \in G_3$ satisfying $u \in E_i$ and $v \in E_j$.

A cyclic expansion simply replaces the loop and vertex corresponding to a multimember equivalence class by a simple cycle through the members of that equivalence class, with each arc between different equivalence classes transformed into a similarly directed single arc between some pair of vertices, one from each of the two equivalence classes.

THEOREM 2. *Given any finite directed graph $G$, let $G_1$ be its equivalent acyclic graph and $G_1^t$ be the unique "transitive reduction" of $G_1$ given by Theorem* 1. *Then the directed graph $H$ satisfies $H^T = G^T$ and has the fewest arcs of any such graph if and only if $H$ is a cyclic expansion of $G_1^t$.*

*Proof.* The proof follows directly from the following lemmas, where $G$, $G_1$, and $G_1^t$ are defined as above.

LEMMA 3. *If $H_1$ is the equivalent acyclic graph for a directed graph $H$ satisfying $H^T = G^T$, then $H_1^T = G_1^T$.*

*Proof.* Since $H^T = G^T$, $H$ contains a path from $u$ to $v$ if and only if $G$ contains a path from $u$ to $v$. Then $H$ and $G$ must have the same vertex equivalence classes, so $H_1$ and $G_1$ are on identical vertex sets. If $H_1^T$ contains an arc $(E_i, E_j)$ not contained in $G_1^T$, then $H_1$ contains a path from $E_i$ to $E_j$ and $G_1$ contains no such path. But then $H$ contains a path from some $u \in E_i$ to some $v \in E_j$ and $G$ contains no such path, a contradiction. Similarly, every arc of $G_1^T$ is an arc of $H_1^T$, so $G_1^T = H_1^T$.

LEMMA 4. *If $H$ has equivalent acyclic graph $H_1$ satisfying $H_1^T = G_1^T$, then $H$ is either a cyclic expansion of $G_1^T$ or $H$ contains more arcs than any cyclic expansion of $G_1^t$.*

*Proof.* If $H_1^T = G_1^T$, we know from Theorem 1 that $H_1$ must contain $G_1^t$. Then $H$ must contain at least one arc for each arc of $H_1$, i.e., if $(E_i, E_j) \in H_1$, there exist $u \in E_i$ and $v \in E_j$ such that $(u, v) \in H$. Furthermore, $H$ must also contain enough arcs to ensure that every pair of vertices belonging to the same equivalence class lie on a cycle in $H$. But this requires at least as many arcs as there are members in the equivalence class, except for single member classes, and such a minimal number of arcs is used if and only if the only arcs between members of the equivalence class form a single cycle including exactly all members of the class. The definition of a cyclic expansion was chosen precisely to include all and only those graphs which use such a minimal number of arcs. Thus, $H$ must either be a cyclic expansion of $G_1^t$ or must have more arcs than any cyclic expansion of $G_1^t$.

LEMMA 5. *Every cyclic expansion of $G_1^t$ has the same number of arcs.*

*Proof.* The number of arcs in any cyclic expansion of $G_1^t$ is exactly equal to the number of arcs in $G_1^t$ plus the number of vertices of $G$ which belong to multimember vertex equivalence classes, minus the number of multimember vertex equivalence classes in $G$.

LEMMA 6. *If $H$ is a cyclic expansion of $G_1^t$, then $H^T = G^T$.*

*Proof.* If $(u, v) \in G^T$, $G$ contains a path from $u$ to $v$. If $u$ and $v$ belong to the same vertex equivalence class, $H$ must contain a path from $u$ to $v$ so $(u, v) \in H^T$.

If $u$ and $v$ belong to different equivalence classes, $u \in E_i$, $v \in E_j$, $G_1$ contains a path from $E_i$ to $E_j$. But then $G_1^t$ contains a path from $E_i$ to $E_j$, so $H$ contains a path from $u$ to $v$ and $(u, v) \in H^T$. Similarly, if $(u, v) \notin G^T$, $G$ contains no path from $u$ to $v$. Also, $u$ and $v$ belong to different vertex equivalence classes, $u \in E_i$, $v \in E_j$, and $G_1$ contains no path from $E_i$ to $E_j$. But then $G_1^t$ contains no path from $E_i$ to $E_j$, and $H$ cannot contain a path from $u$ to $v$, so $(u, v) \notin H^T$.

This completes the proof of Theorem 2.

Theorem 2 tells us that if $G_1$ is the equivalent acyclic graph for $G$, then every cyclic expansion of the graph $G_1^t$ given by Theorem 1 will satisfy our original intuitive definition for a transitive reduction of $G$. In fact, for most algorithms requiring such a transitively reduced graph, the most useful representation will simply be $G_1^t$ along with the corresponding vertex equivalence classes of $G$. However, we also choose to select a unique representative from the various cyclic expansions of $G_1^t$ to be defined as the transitive reduction of $G$.

Let the vertices of $G$ be arbitrarily ordered by assigning them indices as $v_1, v_2, \cdots, v_n$. If $G_1$ is the equivalent acyclic graph for $G$ and $G_1^t$ is the "transitive reduction" of $G_1$ given by Theorem 1, then the *canonical cyclic expansion* of $G_1^t$ is the unique cyclic expansion $G_2$ of $G_1^t$ satisfying:

(i) If $(v_i, v_j) \in G_2$, $v_i \in E_k$, $v_j \in E_k$, and $v_i \neq v_j$, then either $j > i$ and none of $v_{i+1}, \cdots, v_{j-1}$ is in $E_k$ or $v_i$ has the largest index in $E_k$ and $v_j$ has the smallest index in $E_k$; and

(ii) For each arc $(E_i, E_j) \in G_1^t$, $E_i \neq E_j$, there is an arc in $G_2$ from the least index member of $E_i$ to the least index member of $E_j$.

The canonical cyclic expansion merely expands each loop and vertex corresponding to a multimember equivalence class into an ordered simple cycle, with all arcs between equivalence classes transformed into arcs between the least members of the equivalence classes.

We then define the *transitive reduction* of a finite directed graph $G$ to be the unique graph $G^t$ which satisfies:

(i) $(G^t)^T = G^T$;

(ii) If $H^T = G^T$, then $H$ contains at least as many arcs as $G^t$; and

(iii) If $G$ is not acyclic, then $G^t$ is the canonical cyclic expansion of the transitive reduction of the equivalent acyclic graph for $G$.

Existence and uniqueness of $G^t$ follow from the previous results.

We do not attempt a definition of transitive reduction for graphs having infinite vertex sets. However, we point out that additional complications do arise for infinite graphs. Some of these difficulties are illustrated by attempting a reasonable definition of transitive reduction for (i) the countably infinite graph with arcs in both directions between every pair of vertices, and (ii) the infinite graph having a vertex for each real number with an arc from $i$ to $j$ if and only if $i < j$. In neither case does there exist a graph, having the same transitive closure as the given graph, such that no proper subset of that graph also has this property.

**3. Computational complexity of the transitive reduction operation.** We now turn to the question of how quickly the transitive reduction of a graph can be computed. In what follows, we assume that a graph $G$ is represented by its *adjacency*

*matrix*, the matrix with a 1 in row $i$ and column $j$ if there is an arc from the $i$th vertex to the $j$th vertex and a 0 there otherwise. Our results clearly apply to any other graph representation that can be converted to an adjacency matrix and back in time $O(n^2)$, and in which transitive reduction takes $O(n^2)$ time.[1]

We proceed to demonstrate that under the above assumption, the number of steps of a random access computer (e.g., see [4]) needed to compute the transitive reduction of a graph with $n$ vertices differs by at most a constant factor from the time needed to perform Boolean matrix multiplication or to compute the transitive closure of a graph. It should be noted that it was shown in [5] that multiplication of $n \times n$ Boolean matrices requires time which is at most a constant factor more than the time to compute the transitive closure of an $n$ vertex graph, and the converse was shown in [6], [7].

THEOREM 3. *If there is an algorithm to compute the transitive closure of an $n$ vertex graph in time $O(n^\alpha)$, then there is an algorithm to compute transitive reduction in time $O(n^\alpha)$.*

*Proof.* We can compute the transitive reduction of a graph $G$ with $n$ vertices as follows.

1. Find $G_1$, the equivalent acyclic graph of $G$.
2. Let $G_2$ be formed from $G_1$ by deleting loops.
3. Let $M_1$ be the incidence matrix of $G_2$, and let $M_2$ be the incidence matrix of $G_2^T$.
4. Compute $M_3 = M_1 M_2$, and let $G_3$ be the graph whose incidence matrix is $M_3$.
5. Then $G_1^t$ is $G_1 - G_3$.
6. Let $G^t$ be the canonical cyclic expansion of $G_1^t$.

It should be evident that steps 1, 2, 5 and 6 require $O(n^2)$ time. (See [8], e.g., for step 1.) Step 3 requires $O(n^\alpha)$ time to compute $G_2^T$. Step 4 requires $O(n^\alpha)$ time by [4]. Thus, the entire algorithm requires $O(n^\alpha)$ time, since $\alpha \geqq 2$ (see [4]).

It remains to show that $G_1^t = G_1 - G_3$. By Theorem 1, arc $(u, v)$ is in $G_1^t$ if and only if $(u, v) \in G_1$, and there is no path from $u$ to $v$ which does not include arc $(u, v)$. Such a path exists if and only if there is some $w$ not equal to $u$ or $v$ such that there is an arc $(u, w)$ and a path from $w$ to $v$ in $G_1$. These are exactly the conditions under which there will be an arc $(u, v)$ in $G_3$.

THEOREM 4. *If transitive reduction requires $O(n^\alpha)$ steps, $\alpha \geqq 2$, on a graph of $n$ nodes, then transitive closure requires $O(n^\alpha)$ steps.*

*Proof.* Let $G$ be a graph of $n$ vertices. Construct a graph $G'$ with nodes $u$, $u'$ and $u''$ for each vertex $u$ of $G$. The arcs of $G'$ are the following.

1. If $(u, v)$ is in $G$, it is in $G'$.
2. $(u', u)$ and $(u, u'')$ are in $G'$ for each vertex $u$ of $G$.
3. $(u', v'') \in G'$ for all vertices $u$ and $v$ of $G$. $G'$ is shown in Fig. 2.

We observe that $(u', v'')$ is in $(G')^t$ if and only if $(u, v)$ is not in $G^T$. That is, since no arc enters $u'$ or leaves $v''$, both $u'$ and $v''$ are in vertex equivalence classes of their own. By Theorems 1 and 2, $(u', v'')$ is in any transitive reduction of $G'$ if and only if

[1] It may appear that we have defined away the problem. However, a little thought will suffice to conclude that in any reasonable representation, a transitive reduction algorithm will have to "look at" all the arcs and thus take at least time $O(n^2)$.

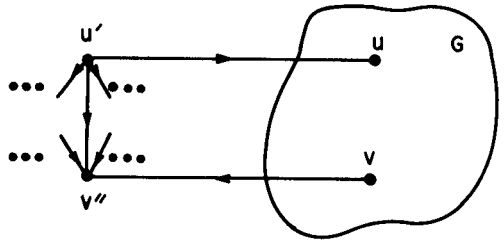


FIG. 2. *Construction of G'*

there is no path of length greater than one from $u'$ to $v''$. But such a path is seen to exist if and only if there is a path from $u$ to $v$ in $G$.

Thus, we may compute $G^T$ by the following algorithm.

1. Construct $G'$.
2. Compute $(G')^t$.
3. Say $(u, v)$ is in $G^T$ if and only if $(u', v'')$ is not in $(G')^t$.

Steps 1 and 3 clearly take time $O(n^2)$ and step 2 requires time $O(n^\alpha)$. Thus, the entire algorithm requires $O(n^\alpha)$ steps.

Theorems 3 and 4 reduce the problem of finding a good algorithm for transitive reduction to that of finding a good algorithm for transitive closure. The method of [6], [7] is based on Strassen's matrix multiplication algorithm [9], and thus takes $O(n^{\log_2 7})$ steps. This method is the best known for large $n$. Under some conditions, transitive closure algorithms found in [10]–[12] may be preferred.

## REFERENCES

[1] E. G. COFFMAN, JR. AND R. L. GRAHAM, *Optimal scheduling for two-processor systems*, to appear.

[2] M. R. GAREY, *Optimal task sequencing with precedence constraints*, to appear in Discrete Mathematics.

[3] D. M. MOYLES AND G. L. THOMPSON, *Finding a minimum equivalent graph of a digraph*, J. Assoc. Comput. Mach., 16 (1969), pp. 455–460.

[4] J. HARTMANIS, *Computational complexity of random access stored program machines*, Rep. TR 70-70, Dept. of Computer Science. Cornell Univ., Ithaca, N.Y., 1970.

[5] M. J. FISCHER AND A. R. MEYER, *Boolean matrix multiplication and transitive closure*, Conference Record, Twelfth Annual Symposium on Switching and Automata Theory, East Lansing, Mich., 1971, pp. 129–131.

[6] I. MUNRO, *Efficient determination of the strongly connected components and the transitive closure of a graph*, unpublished manuscript, Univ. of Toronto, Toronto, Canada, 1971.

[7] M. E. FURMAN, *Application of a method of fast multiplication of matrices in the problem of finding the transitive closure of a graph*, Dokl. Akad. Nauk SSSR, 11 (1970), no. 5, p. 1252.

[8] R. TARJAN, *Depth first search and linear graph algorithms*, Conference Record, Twelfth Annual Symposium on Switching and Automata Theory, East Lansing, Mich., 1971, pp. 114–121.

[9] V. STRASSEN, *Gaussian elimination is not optimal*, Numer. Math., 13 (1969), pp. 354–356.

[10] S. WARSHALL, *A theorem on Boolean matrices*, J. Assoc. Comput. Mach., 9 (1962), pp. 11–12.

[11] V. L. ARLAZAROV, E. A. DINIC, M. A. KRONOD AND I. A. FARADZEV, *On economical construction of the transitive closure of an oriented graph*, Dokl. Akad. Nauk SSSR, 11 (1970), pp. 1209–1210.

[12] P. PURDOM, *A transitive closure algorithm*, BIT, 10 (1970), pp. 76–94.