REFERENCES

[1] M. AJTAI, J. KOMLÓS AND E. SZEMERÉDI, *Sorting in c log n parallel steps*, Combinatorica, 3 (1983), pp. 1–19.

[2] ———, *An O(n log n) sorting network*, Proc. 15th Annual ACM Symposium on Theory of Computing, Boston, 1983, pp. 1–9.

[3] N. ALON, *Eigenvalues and expanders*, preprint, 1984.

[4] N. ALON, Z. GALIL AND V. D. MILMAN, *Better expanders and superconcentrators*, preprint, 1984.

[5] N. ALON AND V. D. MILMAN, *Eigenvalues, expanders and superconcentrators*, Proc. 25th Annual Symposium on Foundations of Computer Science, Gainesville, Florida, 1984, pp. 320–322.

[6] ———, *λ_1, isoperimetric inequalities for graphs, and superconcentrators*, J. Combin. Theory Ser. B, 38 (1985), pp. 73–88.

[7] L. A. BASSALYGO, *Asymptotically optimal switching circuits*, Problemy Peredachi Informatsii, 17(3), (1981), pp. 81–88; Problems Inform. Transmission, 17(3) (1981), pp. 206–211.

[8] L. A. BASSALYGO AND M. S. PINSKER, *Complexity of an optimum non-blocking switching network without reconnections*, Problemy Peredachi Informatsii, 9(1) (1973), pp. 84–87; Problems Inform. Transmission, 9(1) (1973), pp. 64–66.

[9] F. R. K. CHUNG, *On concentrators, superconcentrators, generalizers, and nonblocking networks*, Bell System Tech. J., 58 (1978), pp. 1765–1777.

[10] M. M. DAY, *Convolutions, means, and spectra*, Illinois J. Math., 8 (1964), pp. 100–111.

[11] O. GABBER AND Z. GALIL, *Explicit constructions of linear-sized superconcentrators*, J. Comput. System Sci., 22 (1981), pp. 407–420.

[12] E. HEWITT AND K. ROSS, *Abstract Harmonic Analysis*, Vol. 1, Springer-Verlag, Berlin, Göttingen, Heidelberg, 1963.

[13] H. KESTEN, *Symmetric random walks on groups*, Trans. Amer. Math. Soc., 92 (1959), pp. 336–354.

[14] ———, *Full Banach mean values on countable groups*, Math. Scand., 7 (1959), pp. 146–156.

[15] M. KLAWE, *Limitations on explicit constructions of expanding graphs*, SIAM J. Comput., 13 (1984), pp. 156–166.

[16] G. A. MARGULIS, *Explicit constructions of concentrators*, Problemy Peredachi Informatsii, 9(4) (1973), pp. 71–80; Problems Inform. Transmission, 9(4) (1973), pp. 325–332.

[17] M. S. PINSKER, *On the complexity of a concentrator*, Proc. 7th International Teletraffic Conference, Stockholm, June 1973, pp. 318/1–318/4.

[18] N. PIPPENGER, *Superconcentrators*, SIAM J. Comput., 6 (1977), pp. 298–304.

[19] R. M. TANNER, *Explicit concentrators from generalized N-gons*, this Journal, 5 (1984), pp. 287–293.

SIAM J. ALG. DISC. METH.
Vol. 7, No. 2, April 1986

014

CHARACTERIZATION AND RECOGNITION OF PARTIAL 3-TREES*

STEFAN ARNBORG† AND ANDRZEJ PROSKUROWSKI†‡

Abstract. Our interest in the class of k-trees and their partial graphs and subgraphs is motivated by some practical questions about the reliability of communication networks in the presence of constrained line- and site-failures, and about the complexity of queries in a data base system. We have found a set of confluent graph reductions such that any graph can be reduced to the empty graph if and only if it is a subgraph of a 3-tree. This set of reductions yields a polynomial time algorithm for deciding if a given graph is a partial 3-tree and for finding one of its embeddings in a 3-tree when such an embedding exists. Our result generalizes a previously known recognition algorithm for partial 2-trees (series-parallel graphs).

Key words. graph reductions, confluent reductions, k-trees

AMS(MOS) subject classification. 05C10

1. Introduction. Our interest in the class of k-trees and their subgraphs is motivated by some practical questions about the reliability of communication networks in the presence of constrained line—and site—failures (Farley [5], Farley and Proskurowski [7], Neufeld and Colbourn [10], Wald and Colbourn [14]) and about the complexity of queries in a data base system (Arnborg [1]).

We will briefly describe the connection between the problem of finding the minimal value of k for which a given graph is a partial k-tree and the complexity of queries. Let us consider conjunctive data base queries, an important class of queries from which answers to less restrictive classes of queries can be constructed. Such a query has the form

$$\bigwedge_{1\le i\le k} P_i(x_{i,1}, x_{i,2}, \cdots, x_{i,n_i}).$$

The variables occurring in the relation P_i constitute a subset of n variables $x_1, x_2, \cdots, x_n$; the relation is a set of n_i-tuples of values from a common domain (or from several domains). The query asks if there is an assignment of values $a_1, \cdots, a_n$ to the n variables so that the tuple $\langle a_{i,1}, a_{i,2}, \cdots, a_{i,n_i}\rangle$ belongs to P_i, for each i. The cost of a conjunctive query involving only two relations depends critically on the size of the relations (the number of tuples satisfying them). For a more complex query, minimization of sizes of intermediate relations by means of variable elimination is of great import. This is achieved by answering a partial conjunctive query involving all relations containing a given variable. After this join, the variable can be eliminated from further consideration by simple projection. Joining the relations until only one relation remains evaluates the conjunctive query. If this final relation is nonempty, then the answer is "yes." The size of a k-ary relation may be as large as m^k, where m is the size of the domain. Thus, a relevant objective function for finding the best join order is the maximum arity (the number of variables) of an intermediate relation. Minimization of this objective function is equivalent to embedding a graph obtained from the query syntax into a $(k-1)$-tree, for the minimum k.

In the remainder of § 1 we introduce some standard graph terminology and reduction operations on graphs. In § 2 we review some properties of k-trees and

* Received by the editors January 24, 1984, and in revised form November 15, 1984.

† Department of Numerical Analysis and Computing Science, The Royal Institute of Technology, S-100 44 Stockholm 70, Sweden.

‡ On leave from Computer and Information Science Department, University of Oregon, Eugene, Oregon 97403.

introduce the class of k-decomposable graphs, which is then shown identical to the class of partial graphs of k-trees. In § 3 we exhibit a set of reduction operations such that a graph is reduced to the empty graph by the rules *iff* it is a partial 3-tree. The rules are confluent (or, equivalently, have the Church–Rosser property), which means that the reductions can be applied in any order.

We will consider simple, loopless, undirected combinatorial graphs. Two vertices u and v of a graph G are called *adjacent* if there is an edge (u, v) of G; the edge (u, v) is said to be *incident* with its *end-vertices* u and v. The set of all vertices adjacent to a given vertex v is called the (*open*) *neighborhood* of v in $G, \Gamma_G(v)$ or $\Gamma(v)$ when G is clear from context. The order of $\Gamma(v)$ is called v's *degree*, and its elements are called v's *neighbors*. For a given graph G with the vertex set V and the edge set E, we define a *subgraph induced* by a subset U of vertices to be a graph with the vertex set U and the edge set D of all edges of G with both end-vertices in U. A *clique* is a completely connected subgraph. A *partial graph* of G is defined as a graph with the vertex set V and edge set D, a subset of E. A *subgraph* of G is a partial graph of an induced subgraph of G.

We will investigate classes of graphs which can be defined by the following operators on graphs (see also Rose, Tarjan and Lueker [13]):

Star substitution, $S_k(G, v) = H$, where v is a vertex of G of degree k; the vertex set of H is $V - \{v\}$ and its edge set is $E - \{(u, v) | u \in \Gamma(v)\} \cup \{(u, w) | u, w \in \Gamma(v)\}$. (A star centered in v is "substituted" by a complete graph defined by its neighbors.) This is the vertex elimination operation of Rose [11].

Isolated vertex removal, $I(G, v) = H$, where v is an isolated vertex (with no incident edges); H has the same edge set as G and its vertex set is $V - \{v\}$.

Star removal, $R_k(G, v) = H$, where v is a vertex of G and the subgraph of G induced by $\Gamma(v)$ is a complete graph with k vertices; then $H = S_k(G, v)$.

Star hook-up $H_k(G, K) = H$, where K is a clique induced by k vertices of G; H has the vertex set $V \cup \{w\}, w \notin V$, and the edge set $E \cup \{(u, w) | u$ is a vertex of $K\}$.

Extended operators, S'_k and R'_k, are defined as the unions of I and S_i and R_i respectively, for all i between 1 and k.

We define the corresponding relations $\mathbf{S}_k$, $\mathbf{R}_k$, $\mathbf{H}_k$, $\mathbf{S}'_k$, and $\mathbf{R}'_k$ to hold between two graphs G and H iff there exists an element of G (a vertex or a clique) so that H is the result of applying the corresponding operator to G and its element. Finally, we define the class of k-trees, $\mathcal{T}_k$, (cf. Beineke and Pippert [4] and Rose [11], [12]) as the family of graphs for which the (reflexive) transitive closure of the relation $\mathbf{H}_k$, $\mathbf{H}_k^*$, holds with K_k, the complete graph with k vertices. A *k-leaf* is a vertex of degree k in a k-tree (and in a 3-tree we similarly have 3-leaves). It follows straightforwardly from the definition that a k-tree has at least two k-leaves, and that k-leaves are nonadjacent in a k-tree with more than $k+1$ vertices. The class of *partial k-trees*, $\mathcal{PT}_k$, is defined to consist of all subgraphs of k-trees (cf. Wald and Colbourn [14], for the case of $k = 2$).

It might be interesting to view the reductions S_k as simplifying rewrite rules (this viewpoint has been taken by Liu and Geldmacher [9] who considered the implications of series-parallel reductions). In that context, we would be looking for a set of reduction rules confluent under a congruence relation for which the class of partial k-trees is an equivalence class. Here, the set of reduction rules is confluent if for any two graphs resulting from reducing a given graph in different ways, there exists a reduct graph reachable by reductions from either of the two graphs. Since every chain of reductions of a finite graph is finite, this global confluence property is implied by local confluence, where we require the existence of a common reduct for graphs differing only by one application of different reduction rules (the so-called "diamond lemma," see for instance Huet and Oppen [8]).

2. General k-trees. The following two theorems are equivalent to the example stated in Rose [11] implying that, for a k-tree G, there is a sequence of star removals leading to a complete graph with k vertices, and that any star removal can start this sequence.

THEOREM 2.1. *G is a k-tree iff $G\mathbf{R}_k^*K_k$.*

THEOREM 2.2. *G is a k-tree iff either G is K_k or every H such that $G\mathbf{R}_kH$ is a k-tree.*

The latter gives the basis for an obvious k-tree recognition algorithm (cf. Rose [11]): given a graph G, iteratively remove vertices of degree k with completely adjacent neighbors until no further removal is possible; G is a k-tree iff the remaining graph is the K_k.

After a couple of technical lemmas, we will present a theorem stating that a graph is a partial k-tree if and only if there is a sequence of reductions S'_k which leads to a reduction to the empty graph (with no vertices).

LEMMA 2.3. *For every complete subgraph S of i vertices in a k-tree G $(i < k)$, there exists a complete subgraph K of k vertices in G of which S is an induced subgraph.*

Proof. G can be reduced by a series of applications of operation R_k so that in the resulting k-tree H, no vertex of S is removed and at least one vertex v of S has degree k. In H, the vertex v has a completely connected neighborhood K, or else there would be no way of reducing H to a complete graph through a series of star removals, R_k. K contains all vertices of S. □

LEMMA 2.4. *Any graph with not more than k vertices is a partial k-tree.*

Proof. The complete graph K_k can be constructed by adding, if necessary, new vertices and missing edges to the original graph. □

THEOREM 2.5. *G is a partial k-tree iff $G\mathbf{S}_k'^*\varnothing$.*

Proof. (→, by induction on the order of a k-tree H containing G as a subgraph.) The basis follows by Lemma 2.4 since any vertex v of such a graph G can be chosen for a reduction $S'_k(G, v)$ which, repeated, leads to the empty graph. Let us assume that for any graph G which is a subgraph of a k-tree H with n or less vertices, $G\mathbf{S}_k'^*\varnothing$. Consider a graph G which is a subgraph of a k-tree H with $n+1$ vertices. H has a vertex v of degree k with completely connected neighbors (Rose [11]). If v is a vertex of G, then $G' = S'_k(G, v)$ is a subgraph of $H - \{v\}$; otherwise G is a subgraph of $H - \{v\} = R_k(H, v)$. It follows from the inductive assumption that $G\mathbf{S}_k'^*\varnothing$.

(←, by induction on the order of G.) By Lemma 2.4, we need to consider only graphs with at least k vertices. Let us assume that all graphs with n or fewer vertices which can be reduced to the empty graph by a series of S'_k reductions are partial k-trees. Consider a graph G with $n+1$ vertices and such that $G\mathbf{S}_k'^*\varnothing$. Let v be the vertex of G which is removed in the first of these reductions. Thus, by the inductive assumption, $S'_k(G, v)$ is a subgraph of some k-tree H. By Lemma 2.3, the neighborhood of v is contained in a k-complete subgraph of K and H. Applying the hook-up operation to H and K results in a k-tree containing as a subgraph a graph is isomorphic to G, the new vertex w corresponding to v in G. □

Partial k-trees can be embedded in k-trees without adding any new vertices:

THEOREM 2.6. *Any partial k-tree with at least k vertices can be completed to a k-tree with the same number of vertices.*

Proof (by induction on the number of vertices of G). The theorem is obviously true for G with k vertices. Assume that it is true for all partial k-trees with n vertices and consider a partial k-tree G with $n+1$ vertices. Let v be a vertex of G of degree not greater than k such that the graph $G' = S'_k(G, v)$ in a partial k-tree. There is a k-tree H with n vertices, with G' as a partial graph, and such that a k-complete subgraph K of H contains all vertices of $\Gamma(v)$ (see Lemma 2.3). The k-tree $H_k(H, K)$ has $n+1$ vertices and contains G as its partial graph. □

An alternative definition of partial k-trees can be given using the notion of k-decomposability: a graph G is *k-decomposable iff* either G has $k+1$ or fewer vertices or there is a subgraph S of G with at most k vertices such that $G-S$ is disconnected, and each of the connected components of $G-S$ augmented by S with completely connected vertices is k-decomposable.

THEOREM 2.7. *The class of k-decomposable graphs is exactly $\mathcal{PT}_k$.*

Proof. Since all minimal separators in a k-tree have order k (Beineke and Pippert [4]), k-trees are k-decomposable, together with all their partial graphs. If a k-decomposable graph with more than $k+1$ vertices can be embedded in a union of two k-trees with at most k completely connected vertices in common, then—by Lemma 2.3—the common part can be extended to a k-complete graph, the embedding graph can be extended to a k-tree, and the theorem follows by induction on the order of the graph. □

Unfortunately, this characterization of partial k-trees does not give us an efficient algorithm for recognition of this class of graphs since the separator property is lost in subgraphs. Namely, in a partial k-tree we may be able to find a "small" separator which cannot be extended to a complete subgraph of a k-tree in an embedding of the partial k-tree.

3. Partial 3-trees. Wald and Colbourn [14] restate Duffin's [5] characterization of series-parallel graphs by completely characterizing the class of partial 2-trees as graphs with no subgraphs homeomorphic to K_4. This characterization does not carry into higher values of k. Figure 1(a) shows a planar graph (which cannot have a homeomorph of K_5) which is not a partial 3-tree.

A natural generalization of the recognition algorithm for series-parallel graphs [5], [10] would be to perform applicable reductions in S_3' in any sequence. Since the operations I and S_1 do not introduce any new edges, they result in partial 3-trees whenever applied to a partial 3-tree. However, the other reduction operations in S_3' may not be "safe," i.e., a partial 3-tree may be reduced to a graph which is not a partial 3-tree. An example is given in Fig. 1(b), where a partial 3-tree can be reduced to the graph in Fig. 1(a) by application of S_3 to vertex v. We can recognize a simple case of safe application of the reduction S_3':

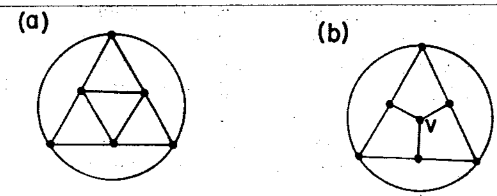

FIG. 1. (a) *The 6-vertex, 4-regular plane graph G, and* (b) *a partial 3-tree H such that $G=S_3(H, v)$.*

THEOREM 3.1. *For any partial 3-tree G, $S_2(G, w)$ is a partial 3-tree and $S_3(G, v)$ is a partial 3-tree if at least two neighbors of v are adjacent. See Fig. 2.*

Proof (by induction on the number of vertices in G). The theorem is obviously true if G has not more than 3 vertices. Assume that in any partial 3-tree with less than $n>3$ vertices such reductions lead to partial 3-trees. Consider a partial 3-tree G with n vertices and a reduction $G_3'(G, u)$ resulting in a partial 3-tree, G'. If $u=w$, $u=v$, or u is not any of the neighbors of v or w, then the theorem follows directly from the assumption. If degree of u is less than 3, the cases of its adjacencies are trivial. Otherwise, there are three cases to consider: (i) u is a neighbor of w. Then, $S_3(G', w)$

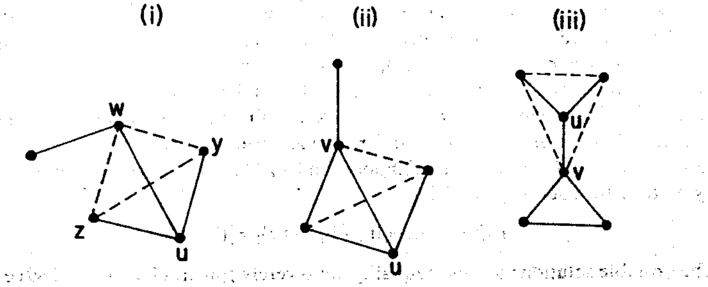

FIG. 2. *Cases in the proof of Theorem* 3.1.

is a partial 3-tree by the inductive assumption (two of w's neighbors in G', y and z, are adjacent) and it is isomorphic to $S_3(S_2(G, w), u)$. (ii) u is one of the adjacent neighbors of v. If u is also adjacent to the third neighbor of v, then G' is isomorphic to $S_3(G, v)$. Otherwise, $S_3(G', v)$ is a partial 3-tree by the inductive assumption (v's neighbors in G', x and y, are adjacent) and it is isomorphic to $S_3(S_3(G, v), u)$. (iii) u is not adjacent to the other two neighbors of v. Then, $S_3(G, v)$ is isomorphic to a partial graph of $S_3(G, u)$, which is a partial 3-tree by the assumption. This completes the proof of the inductive step. □

The operation S_3 plays a crucial role in the eventual reduction of vertex degrees in the graph, even though the star substitution operation applied to vertices with independent (nonadjacent) neighbors increases their degree. Fortunately, we are able to isolate configurations involving such independent neighborhoods of vertices of degree 3 that make the degree reduction possible.

THEOREM 3.2. *A graph G without vertices of degree* 0, 1, *or* 2, *and with no vertex of degree* 3 *that has two adjacent neighbors is a partial* 3*-tree only if there are subgraphs of G isomorphic to either C' or C'' in Fig.* 3, *where vertices u, v and w have degree* 3 *in G, and vertex x of C' has degree* 3 *in G.*

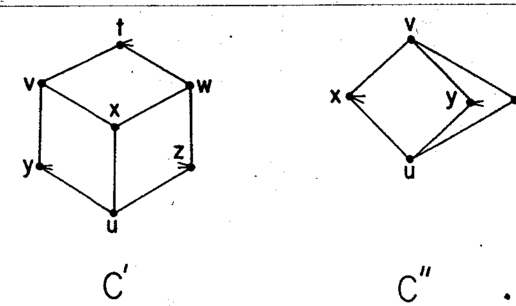

FIG. 3. *The necessary subgraphs in a partial 3-tree.*

Proof. Consider a partial 3-tree G such that its minimum vertex degree is 3 and no two neighbors of a degree 3 vertex are adjacent. Let H be one of G's embedding 3-trees. H' is the induced subgraph of H obtained by deleting all 3-leaves from H (at least two exist). G' is the graph obtained from G by removing the 3-leaves with the S_3 reduction. G' is a partial graph of H'. Let x be any 3-leaf of H' (or any vertex if

$H' = K_4$), and let L_x be the nonempty set of 3-leaves of H adjacent to x. Every member of L_x has degree 3 in H and thus degree 3 or less in G which is a partial graph of H. But G has also minimum degree 3, so each member of L_x has degree 3 in G. Since no two neighbors of a degree 3 vertex are adjacent in G, no two neighbors of a vertex in L_x are adjacent in G, so L_x and x have disjoint neighborhoods in G. The degree of x in G' is not greater than 3 and x's neighborhood in G' consists of two disjoint sets, $\Gamma_G(x) - L_x$ (the original neighbors) and $\Gamma_G(L_x) - \{x\}$ (the neighbors introduced by S_3 transformations), i.e.:

$$|\Gamma_G(x) - L_x| + |\Gamma_G(L_x) - \{x\}| = |\Gamma_{G'}(x)| \leqq 3.$$

The possible solutions to this inequality are severely constrained by the degree assumption on G and the fact that for both set differences the second operand is a subset of the first. Since G has minimum vertex degree 3, the second term in the left-hand side is at least 2, so the first term can only be 0 or 1, and $|L_x|$ is at least two. We split cases by the second term, which can be 2 or 3:

(i) $|\Gamma_G(L_x)| = 3$. All vertices in L_x have the same neighbors, and since they are at least two, configuration C'' must be present.

(ii) $|\Gamma_G(L_x)| = 4$. This case implies $\Gamma_G(x) = L_x$, so $|L_x|$ must be at least 3, and every vertex in L_x is G-adjacent to x and two other vertices among the three in $\Gamma_G(L_x) - \{x\}$. This implies configuration C'' if $|L_x| \geqq 4$ and C' or C'' if $|L_x| = 3$. Since $\Gamma_G(x) = L_x$, whenever only configuration C' is present, one of its occurrences must have degree 3 in G for its vertex x. □

Now that we know of the necessity of subgraphs with vertices of degree 3 involved in triangles or squares, we have to establish safe reductions thereof. For example, in the graph G (see Fig. 4) which has the graph C' of Fig. 3 as a subgraph, reduction $S_3(G, x)$ leads to a graph which is not a partial 3-tree even though the original graph G is one.

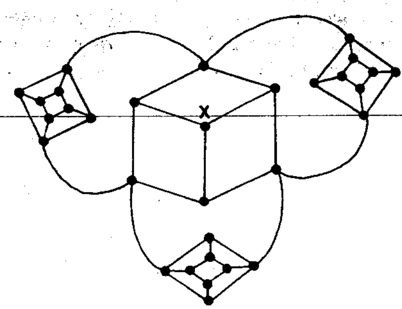

FIG. 4. *A partial 3-tree G with an unsafe reduction $S_3(G, x)$.*

THEOREM 3.3. *For any partial 3-tree G with a subgraph isomorphic to either graph C' or C'' in Fig. 3, the graphs $S_3(G, u)$, $S_3(G, v)$, and $S_3(G, w)$ are all partial 3-trees if vertices u, v, and w have all degree 3 in G.*

Proof (by induction on the number of vertices of G). By inspection, the thesis is true for graphs with no more than 6 vertices. Assume it is true for graphs with fewer than $n > 6$ vertices and consider a partial 3-tree G with n vertices. By Theorem 2.5, there is a vertex s such that $S_3'(G, s)$ is a partial 3-tree. If s is one of the vertices u, v or w, or is not adjacent to any of them, then the thesis follows by the inductive assumption. Otherwise, we have to consider three cases: (i) In a subgraph isomorphic to C', $s = x$. Applying the operation S_3 to u, v, w and x, in this order, reduces graph G to a graph G_1, where the three remaining vertices induce a triangle (see Fig. 5(a)). Since the graph $S_3(G, s)$ is a partial 3-tree, so is its subgraph G_2 in which the edges (u, v), (v, w), and (u, w) are missing. But G_2 is reducible to G_1 by application of the operation S_2 (which is safe by Theorem 3.1) to u, v, and w (see Fig. 5(b)). Thus, $S_3(G, u)$ is a partial 3-tree.

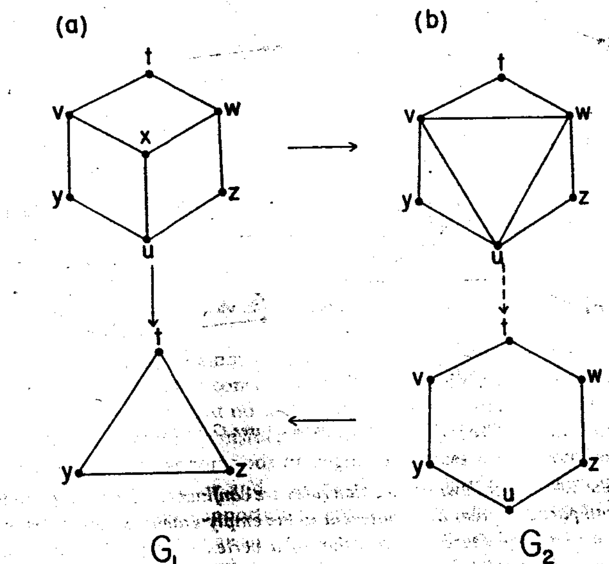

FIG. 5. *Reduction of a partial 3-tree G, case (i).*

(ii) In a subgraph isomorphic to C', $s = y$ (by symmetry, a similar argument holds for $s = z$ and $s = t$). Applying the operation S_3 to vertices u, v, and w in this order reduces G to a graph containing as a subgraph the graph G_1 (see Fig. 6(a)). On the other hand, since $S_3(G, y)$ is a partial 3-tree, so is its subgraph G_2 in which the edge (q, v) is missing (q is the third neighbor of y). But G_2 is reducible to a graph isomorphic to G_1 (in which u is isomorphic to y in G_1) by a sequence of safe applications of the operation S_3 to v and w (see Fig. 6(b) and Theorem 3.1). Thus, $S_3(G, u)$ is a partial 3-tree.

(iii) In a subgraph isomorphic to C'', $s = x$ (by symmetry, a similar argument applies to the cases $s = y$ or $s = z$). Applying the operation S_3 to u, v, and x in this order reduces G to a graph G_1 where the remaining vertices induce a triangle (see Fig. 7(a)). The graph $G_2 = S_3(G, s) - \{(v, q)\}$ (where q is the third neighbor of x) is a partial 3-tree and is reducible to G_1 by application of the safe instances of the operation S_3 to v and u in this order (see Fig. 7(b)). Thus G_1 is a partial 3-tree. □

It should be obvious that the reduction S_2' is confluent in a system recognizing partial 3-trees, since $S_2'(S_2'(G, u), v) = S_2'(S_2'(G, v), u)$, for any graph G and its two vertices u and v. By inspection of cases when two adjacent vertices can be reduced according to any two safe reduction rules, we can easily show that the set of instances of S_3' investigated in Theorems 3.1 and 3.3 are confluent.

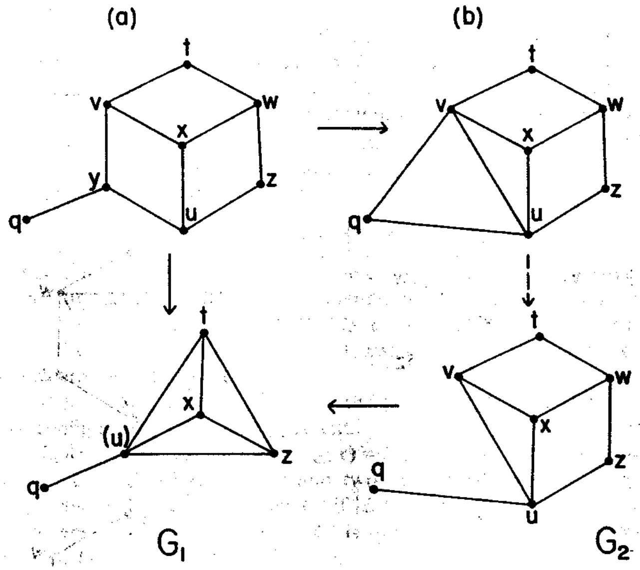

FIG. 6. *Reduction of a partial 3-tree G, case* (ii).

THEOREM 3.4. *The following reduction rules are confluent under a congruence relation under which all partial 3-trees are equivalent to the empty graph: isolated vertex removal, reduction of a vertex of degree* 1, *reduction of a vertex of degree* 2, *and star-triangle substitution when* (i) *two of the neighbors of a vertex of degree* 3 *are adjacent* (*Theorem* 3.1), (ii) *all the neighbors of a vertex of degree* 3 *are also neighbors of one other vertex of degree* 3, *or* (iii) *the neighbors of a vertex of degree* 3 *are shared with those of two other vertices of degree* 3 *that also share a fourth vertex* (*Theorem* 3.3).

The theorem above has a generalization which concerns a complete set of reduction rules. A set of reduction rules will be called *complete* if they are sufficient to reduce all and only graphs from a given class to a given canonical form. In this context, a reduction rule is safe if it cannot take a member of the class outside the class.

THEOREM 3.5. *Each of the reductions from a complete set of confluent rules is safe.*

Proof. Let us assume, to the contrary, that a graph x in a class C has a successor (reduct) y not in C. Since the reduction rules are complete, x can be reduced to the canonical form z, while y can not. However, confluence implies that all reducts of x have a common successor. □

The results of this section yield an $O(n^3)$ algorithm for finding an embedding 3-tree of a graph with n vertices or deciding that no such embedding exists. The time required for performing the S'_3 reductions once the vertex order has been decided is, with suitable data structures (see Wald and Colbourn [14]) $O(n)$. In order to find the next vertex to reduce, first in time $O(n)$ select a vertex of degree $\leqq 2$ or a vertex of degree 3 with either (i) two adjacent neighbors (in which case the selected vertex is reduced) or (ii) three neighbors of degree 3 with overlapping neighborhoods so that the selected vertex is x in a subgraph isomorphic to C' of Fig. 3 (in this case the neighbors of x are reduced). If no such vertex exists, check for all $O(n^2)$ degree 3

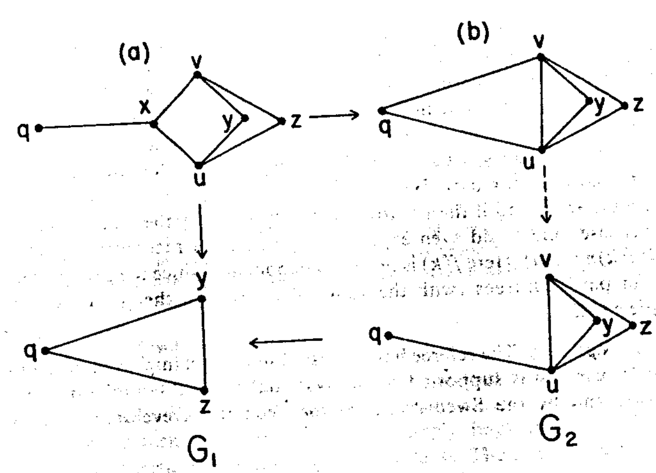

FIG. 7. *Reduction of a partial 3-tree G, case* (iii).

vertex pairs if they have common neighborhood (in which case they are reduced). Each pair can be processed in constant time. The total worst-case time for finding the reduction order or deciding that no such order exists is clearly $O(n^3)$. Our referee has pointed out that this can be considerably improved by two modifications: The first consists in having all neighborhoods of degree 3 vertices (where vertices have been numbered in an arbitrary order and each neighborhood is represented as an ordered triple) as keys in a structure supporting insert, search and delete in time $O(\log n)$ (e.g. an AVL tree). This makes it unnecessary to examine all pairs in order to find configuration C''. The second improvement consists in having a list of ready vertices (that fulfill the conditions for safe reduction). Each reduction made implies a neighborhood change for at most 3 vertices and removal of at most 4 vertices from the ready list (which means that the total number of additions and deletions in the ready list is $O(n)$). With these improvements worst-case processing times of $O(n \log n)$ seem possible.

4. Conclusion and further research. We have found a set of confluent reductions on graphs such that any graph can be reduced to the empty graph if and only if it is a partial 3-tree. This set of reductions yields a polynomial time algorithm for recognizing partial 3-trees and embedding them in full 3-trees. This generalizes the previously known recognition algorithm for partial 2-trees of Wald and Colbourn's [14].

Already for the case of $k=4$, there is no easy generalization of our methods used in recognition of partial 3-trees. A solution to this problem for arbitrary k would have significant practical applications, since graph algorithms based on decomposition are frequently used, even though only heuristic decomposition strategies are known. The cost of such a decomposition algorithm is often exponential in the order of the articulation sets used. Thus, the minimax solution given by a k-tree embedding for the minimum value of k is clearly highly relevant.

In a preliminary presentation of this research we describe families of safe (but not necessarily complete) reductions for general partial k-trees [2, Thms. 4.1–4.6]. We have programmed these reduction rules and tested them on partial k-trees generated by Monte-Carlo techniques. For small values of k (up to 7), almost all of the graphs were correctly recognized, but the failure rate grew with increasing k. The existing, incomplete set of safe reduction rules could thus be used as another heuristic decomposition method. It differs from most such methods in its "bottom-up" (rather than "top-down") approach. This method has the worst case complexity of order $O(n^k)$, which compares favorably with the $O(n^{k+2})$ complexity of the only known complete recognition algorithm for partial k-trees [3]. If complete sets of safe reductions are found for arbitrary k and if the improvements suggested by the referee (see § 4) carry over to this case, one could even expect an algorithm for recognizing partial k-trees in time $O(f(k)n \log n)$. Here $f(k)$ is probably exponential since the general recognition problem for partial k-trees (with the value of k given in the problem instance) is NP-complete [3].

Acknowledgments. The referee has suggested significant improvements and corrections. This research was supported by the National Science Foundation under grant INT-8318441 and by the Swedish Board for Technical Development under grant 83-3719.

REFERENCES

[1] S. ARNBORG, *On the complexity of multivariable query evaluation*, FOA Rapport C20292-D8, National Defence Research Institute, Stockholm, Sweden, 1979.

[2] S. ARNBORG AND A. PROSKUROWSKI, *Characterization and recognition of partial k-trees*, TRITA-NA 8402, Royal Institute of Technology, Sweden, 1984.

[3] S. ARNBORG, D. G. CORNEIL AND A. PROSKUROWSKI, *Complexity of finding embeddings in a k-tree*, TRITA-NA 8407, Royal Institute of Technology, Sweden, 1984.

[4] L. W. BEINEKE AND R. E. PIPPERT, *Properties and characterizations of k-trees*, Mathematika, 18 (1971), pp. 141–151.

[5] R. J. DUFFIN, *Topology of series-parallel networks*, J. Math. Anal. Appl., 10 (1965), pp. 303–318.

[6] A. M. FARLEY, *Networks immune to isolated failures*, Networks, 11 (1981), pp. 255–268.

[7] A. M. FARLEY AND A. PROSKUROWSKI, *Networks immune to isolated line failures*, Networks, 12 (1982), pp. 393–403.

[8] G. HUET AND D. OPPEN, *Equations and rewrite rules: a survey*, in Formal Languages: Perspective and Open Problems, R. Book, ed., Academic Press, New York, 1980.

[9] P. C. LIU AND R. C. GELDMACHER, *Graph reducibility*, in Proc. Seventh S-E Conf. Combinatorics, Graph Theory, and Computing, Utilitas Mathematica, Winnipeg, 1976, pp. 433–455.

[10] E. M. NEUFELD AND C. J. COLBOURN, *The most reliable series-parallel networks*, TR 83-7, Dept. Comp. Sci., Univ. Saskatchewan, 1983.

[11] D. ROSE, *Triangulated graphs and the elimination process*, J. Math. Anal. Appl., 32 (1970), pp. 597–609.

[12] ———, *On simple characterization of k-trees*, Discrete Math., 7 (1974), pp. 317–322.

[13] D. ROSE, R. E. TARJAN AND G. S. LUEKER, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.

[14] J. A. WALD AND C. J. COLBOURN, *Steiner trees, partial 2-trees, and minimum IFI networks*, Networks, 13 (1983), pp. 159–167.

SIAM J. ALG. DISC. METH.
Vol. 7, No. 2, April 1986

015

AN APPLICATION OF THE SINGULAR VALUE DECOMPOSITION TO MANIPULABILITY AND SENSITIVITY OF INDUSTRIAL ROBOTS*

MASAKI TOGAI†

Abstract. In designing and evaluating industrial robots, it is important to find optimal configurations and locate optimum points in the workspace for the anticipated tasks. In the current paper the singular value decomposition and perturbation analysis are applied to the Jacobian of robot kinematics; the condition number of the Jacobian is then proposed to be a measure of the "nearness" to degeneracy. Then qualitative measures called kinematic "manipulability" and "sensitivity" are proposed. Some properties of proposed measures are investigated and the relation between these measures are discussed. Optimal postures of various types of industrial robots are obtained.

AMS(MOS) subject classifications. 15, 65

1. Introduction. In designing and evaluating industrial robots, it is important to find optimum configurations, or postures, and locate optimum points in the workspace for the anticipated tasks. This becomes increasingly important in high precision assembly. Several measures of workspace are available. The size of reachable volume is an important performance measure [1]. To obtain full mobility throughout its range of motion, the ideal manipulator would have no singularities, or degeneracies, in its workspace. In general, a manipulator becomes degenerate in its workspace; therefore, the "nearness" to the degeneracy is also an important measure. Yoshikawa [2], [3] calls it "manipulability." Another important measure of workspace quality is the accuracy with which the task would be achieved. Particularly if the magnitude of accuracy of the manipulator is comparable to that of the anticipated tasks such as high precision assembly tasks, this measure is extremely important.

In this paper a new qualitative measure for manipulability is proposed. Advantages of the proposed measure over Yoshikawa's definition are discussed. Another qualitative measure of a manipulator's ability of accurately positioning and orienting a manipulator, so-called *sensitivity*, is proposed. Some properties of proposed manipulability and sensitivity are investigated and the relation between these measures are discussed. Optimal postures of various types of manipulators are obtained. Some computational consideration of proposed manipulability and sensitivity are also discussed.

2. Manipulability: A new definition. Yoshikawa [2] proposed $w = \sqrt{\det(JJ^T)}$ for a qualitative measure of manipulating ability of robot arms, and called it *manipulability*. According to the singular valued decomposition (SVD) theorem [4], assuming J is m-by-n matrix there exist orthogonal matrices $U \in R^{m\times m}$ and $V \in R^{n\times n}$ such that

(1) $$J = U \Sigma V^T,$$

where

$$\Sigma = \left[\begin{array}{cccc|c} \sigma_1 & & 0 & & \\ & \sigma_2 & & & 0 \\ 0 & & \ddots & & \\ & & & \sigma_m & \end{array}\right] \in R^{m\times n}.$$

* Received by the editors January 15, 1985. This paper was presented at Second SIAM Conference on Applied Linear Algebra, Raleigh, North Carolina, April 29–May 2, 1985.

† AT&T Bell Laboratories, Holmdel, New Jersey 07733.