

Short-circuited k -trees

By I. PÁVÓ

Abstract

In this paper the author introduces the idea of the short-circuited k -tree arising from k -tree of a graph in each component of which a vertex is given in advance. Necessary and sufficient condition is given for short-circuited k -trees to be circuitless and a procedure to generate is shown, too. During this procedure the generation of k -trees published by the author in an earlier paper [6] is used as well. The authors methods can be applied in designing of general linear electrical networks by topological formulas.

Introduction

During topological design of linear active networks one has to discover k -trees of the graph of electrical network representing common trees in the network reduced in two different ways [4]. More exactly the task is to discover such k -trees of the graph of network which represent connected circuitless subgraphs either short-circuiting all the nullators [7] and deleting the norators or short-circuiting all the norators [7] and deleting the nullators.

The application of topological formulas is very difficult in this manner and it can be considered as a solved problem only in principle [3].

But there is another possibility to design linear active networks by topological formulas which is more congenial also to computer science. Namely the suitable k -trees can be selected from a set of k -trees which was generated by method [6] based on theorem of Ore [5]. Each component of these k -trees has the common property to contain exactly one of the selected vertices from the graph [6]. k -trees with such property may be advantageously used in design of passive networks as well. The suitable k -trees can be produced from the above set by selecting the graphs being circuitless after fusion of certain pairs of vertices in each k -tree of the set.

In the present paper we are going to deal with "short-circuited" in one or more pairs of vertices k -trees being generated by method [6] from the graph of network in the form $v(\mu^{-1}(\mathbf{M}_{i_1, \dots, i_k}))$, where G is the graph of network, $\mu(G)$ is its adjacency matrix, μ^{-1} is the inverse of μ , $\mathbf{M}_{i_1, \dots, i_k}$ is an (i_1, \dots, i_k) -reduction of $\mu(G)$, and v is the operator removing the direction.

Short-circuited k -tree and its properties

Consider a simple graph G (i.e. an undirected graph without loops and multiple edges) with vertices P_1, \dots, P_n . Let us select arbitrarily k different vertices ($k \leq n$) and construct an F_{i_1, \dots, i_k}^k k -tree of G (we can use the method [6]), where i_1, \dots, i_k are indices of the selected vertices. Then choose N pairs of vertices of G pair-wise different at least in one vertex.

Definition. A short-circuited in N pairs of vertices k -tree is a graph arising from F_{i_1, \dots, i_k}^k by considering each chosen pair of vertices one (or in other words each chosen pairs of vertices is separately short-circuited).

Thinking of the directed graph $\mu^{-1}(\mathbf{M}_{i_1, \dots, i_k})$ we can supply the short-circuited k -tree with unambiguous direction if we direct the edges of the corresponding short-circuited k -tree in the same way. So we get the directed short-circuited in N pairs of vertices k -tree. Further on we shall speak shortly about directed or undirected short-circuited k -trees.

We mention some properties of a short-circuited k -tree following from its definition:

(i) The number of its vertices is $n - N$, the number of its edges is equal to the number of the edges of the suitable k -tree, and the maximal number of its components is k .

(ii) The components of a short-circuited k -tree may generally contain circuits or loops.

(iii) The circuits of an undirected short-circuited k -tree will not be always directed circuits of the corresponding directed one.

(iv) The vertices of the short-circuited k -tree either correspond to the original ones of the k -tree or they have been arisen by fusion. Vertices arisen by fusion are called multiple vertices, vertices with indices i_1, \dots, i_k are the selected vertices and the remaining ones are the usual vertices.

(v) To a vertex of the directed short-circuited k -tree more than one edge directed outwards can incident. But such a vertex can be only a multiple one. Outwards directed edges are incident to a multiple vertex if and only if they were arisen by fusion of usual vertices.

(vi) If there is a circuit (or a loop) in a short-circuited k -tree, then it has at least one multiple vertex. Otherwise passing back to the corresponding k -tree, it also contains circuit (or loop) contradicting the definition of k -tree.

Now we are taking an interest in the necessary and sufficient condition for an undirected short-circuit k -tree to contain circuit (or loop).

(vii) A short-circuited k -tree contains a circuit (or a loop) if and only if different vertices P_1, \dots, P_l can be chosen from its multiple vertices so that between arbitrary two adjoining vertices of the sequence P_1, \dots, P_l, P_1 there exists either an arc progression or a non-multiple vertex connected by a path progression with its adjoining multiple vertices in the corresponding directed short-circuited k -tree.

To verify (vii) we remark that the vertices P_1, \dots, P_l in question are in order all the multiple ones of a circuit (or a loop) of the short-circuited k -tree (see vi); it is obvious that two arbitrary adjoining vertices of the sequence P_1, \dots, P_l, P_1 are connected by chain. Vertices of this chain are generally different, not more than the first and the last vertices may be equal. This is the situation when $l=1$, otherwise the chain between two multiple adjoining vertices is a path. Because of the

construction of the directed k -tree this is an arc progression in the corresponding short-circuited k -tree, if no selected vertex occurs between the middle vertices. Otherwise only one selected vertex may exist between the middle vertices (see v), which is really connected by path progression with its adjoining multiple vertices.

Cycle check on short-circuited k -trees

In what has gone before we examined the necessary and sufficient condition for a short-circuited k -tree to have circuit. Now we are going to study the following problem:

Construct all the F_{i_1, \dots, i_k}^k k -trees of a simple graph. Fuse one by one the same N pairs of vertices in each of the k -trees. Among the short-circuited k -trees may be graphs which contain circuit (or loop) but circuitless graphs can also occur. The latter are obviously $(k-N)$ -trees. Now the problem is how can we choose the $(k-N)$ -trees from the short-circuited k -trees, or, what is an equivalent problem, how to recognize the graphs containing circuit (or loop).

We have met a similar problem in the discussion of the generation of k -trees (see [6]). There we could choose k -trees from the set of the generalized trees by complete cycle check. To solve the present problem we will apply the earlier procedure. First of all we characterize a short-circuited k -tree by row vector representation.

Definition. Row vector representation of a short-circuited in N pairs of vertices k -tree is called the sequence (s_1, \dots, s_n) with n members which arises from the row vector representation of the original k -tree after the indices of the fused vertices are marked by a common symbol in order of succession of the N pairs of vertices.

It is obvious that the introduction of common symbols may be performed by choosing of a common symbol to the natural number indices of each pair of vertices one by one. Therefore the common symbol can be considered as an index of the corresponding multiple vertex.

Completed row vector representation of a short-circuited k -tree is called a matrix of size $2 \times n$ the first row of which consists of the natural numbers $1, 2, \dots, n$, taking the introduced common symbols into account, and its second row is the row vector representation of the short-circuited k -tree.

Observe that from a completed row vector representation we can easily pass to the actual directed short-circuited k -tree. Disregarding the columns of matrix containing zero in their second row, the remaining columns indicate the pair of vertices that are connected in the directed graph in question. Therefore the completed row vector representation is called as the representation of the short-circuited k -tree as well.

We remark that the first row in the representation of a short-circuited k -trees can contain of the natural number $1, 2, \dots, n$ item symbol-elements, while among the element of the second row can occur the number 0. In the first row the number $1, 2, \dots, n$ can occur no more that one time, while the symbol elements can do it several times, too. Finally notice that the number of the same symbol elements shows the number of the edges being incident with the corresponding multiple-vertex and that are directed outwards.

Definition. A function $\varphi(x)$ is called the function associated with a completed row vector representation the domain of which is the set of the elements of the first row, its range is the set of the elements of the second row and the correspondence is defined by the column of the representation in question. Generally $\varphi(x)$ is a function of multivalued.

Remember that there was a "function $\varphi(x)$ " in the previous paper [6] as well. The idea of the cycle check was just constructed by application of that function. Presently we introduce the function $\varphi(x)$ with similar design.

Definition. Let be $\varphi(x)$ the associated function with the completed row vector representation

$$\mathbf{R} = \begin{pmatrix} r_1 & \dots & r_n \\ s_1 & \dots & s_n \end{pmatrix}.$$

By a cycle check performed on \mathbf{R} starting with the element r_i we mean the construction of the sequence

$$r_i, \varphi(r_i), \varphi(\varphi(r_i)), \dots \quad (1 \leq i \leq n).$$

We say that the outcome of the cycle check is finite if we can construct only a finite sequence, i.e. if somewhere in the sequence a zero turns up, which does not belong to the domain φ ; otherwise we say that the outcome of the cycle check is infinite.

It the outcome of the cycle check is infinite then, as it can be easily seen, from a certain point the same segment of the above sequence will occur repeatedly.

A fundamental difference appears between "the present cycle check" and the earlier one published in the paper [6]. The cycle check performed on \mathbf{R} starting at r_i can not be unambiguous therefore the outcome can be several. Such a cycle check can arise if among the members of the sequence performed to the cycle check a symbol element does occur for the function φ is generally of multivalued on symbol elements.

Moreover the construction of the sequence $r_i, \varphi(r_i), \varphi(\varphi(r_i)), \dots$ can also be regarded as walking through a part of the graph, starting at a vertex P_{r_i} of the directed short-circuited k -tree and always proceeding in conformity with the direction of the edges passed along. It goes without saying that if we start cycle checks with each r_i than we walk the whole graph (generally several times) and in case of infinite outcome we get into a directed circuit (or loop) during the walk. But to find out the existence of a circuit (or a loop) it is sufficient if we start cycle checks only with symbol elements namely because of (vi) a circuit (or a loop) always has multiple vertex and symbol elements are just the indices of the multiple vertices.

Definition. By a complete cycle check performed on a completed row vector representation we mean a bunch of all possible cycle checks starting with all symbol elements of the first row of the representation in question. The outcome of a complete cycle check is said to be finite if all checks constituting it have finite outcomes; otherwise, the outcome is said to be infinite.

Because of "the graph theory background" of the cycle check later we shall say that the cycle check is performed on the short-circuited k -tree, in particular starting at its P_{r_i} vertex, by which we mean that the cycle check is performed on the representation starting at the symbol r_i . The idea of the complete cycle check performed on a short-circuited k -tree will be used in similar meaning as well.

Short-circuited k -trees without circuit (or loop)

Consider a short-circuited in N pairs of vertices k -tree with P_{a_1}, \dots, P_{a_N} multiple vertices. Let be i_1, \dots, i_k the indices of the selected vertices in the corresponding k -tree. Suppose that the complete cycle check performed on the short-circuited k -tree has a finite outcome. It means that all possible cycle checks starting at all multiple vertices come to an end somewhere. Tabulate about the finish of each possible cycle check, that is put down in order those indices of the columns of the completed row vector representation in which the cycle checks have finished together with indicating the vertices where checks were started from. It is obviously enough if the table contains only the indices of the vertices in question. So we get the table:

a_1	...	a_j	...	a_N
$i_1^1, \dots, i_{m_1}^1$...	$i_1^j, \dots, i_{m_j}^j$...	$i_1^N, \dots, i_{m_N}^N$

The meaning of the above table is the following: Cycle checks starting at P_{a_j} can be performed exactly of number m_j which finish at vertices $P_{i_1^j}, \dots, P_{i_{m_j}^j}$ in order, where $j=1, \dots, N$. Naturally among the numbers $i_1^j, \dots, i_{m_j}^j$ equal elements may occur as well. Notice that the obtained elements $i_1^j, \dots, i_{m_j}^j$ where the cycle checks finished can be considered as indices of selected vertices. Namely if the superior element of the column of the representation where the cycle check was finished is not a symbol element then during the cycle check the last touched vertex is just a selected one. If the superior element was a symbol one then the corresponding vertex had arisen with short-circuiting of some selected vertex of the original k -tree.

Definition. A graph with vertices $P_{a_j}, P_{i_1^j}, \dots, P_{i_{m_j}^j}$, and with edges $(P_{a_j}, P_{i_1^j}), \dots, (P_{a_j}, P_{i_{m_j}^j})$ ($j=1, \dots, N$) is called the reduced graph of the short-circuited k -tree. It is always undirected.

Notice that the idea of the reduced graph is defined only in that case if the outcome of the complete cycle check is finite. Otherwise the reduced graph is generally more simple as the original one which was reduced and it is a bipartite graph [1].

Theorem. A short-circuited k -tree is without a circuit (or a loop) if and only if the complete cycle check performed on it has a finite outcome and its reduced graph is circuitless.

Proof. To verify the sufficient condition assume that the complete cycle check performed on the short-circuited k -tree has a finite outcome, the reduced graph is circuitless nevertheless the short-circuited k -tree contains circuit (or loop). Then the corresponding subgraph of this circuit (or loop) cannot be a directed one in the directed short-circuited k -tree. Let be all the multiple vertices P_{a_1}, \dots, P_{a_l} in order that are incident with the circuit (or loop). According to (vii) cycle checks starting at adjoining multiple vertices finish at a place of common column index in the corresponding representation where P_{a_l} is adjoining P_{a_1} , too. Let be P_{i_j} the selected vertex defined by the common column index belonging to the cycle checks starting at P_{a_j} and $P_{a_{j+1}}$ ($j=1, \dots, l; l+1=1$). So edges $(P_{a_1}, P_{i_1}), \dots, (P_{a_l}, P_{i_l})$ determine a circuit in the reduced graph contradicting to the starting assumption.

The condition is necessary, too. Namely if the complete cycle check performed on a short-circuited k -tree has an infinite outcome then the corresponding directed short-circuited k -tree contains a circuit (or a loop) and for the same reason so does the

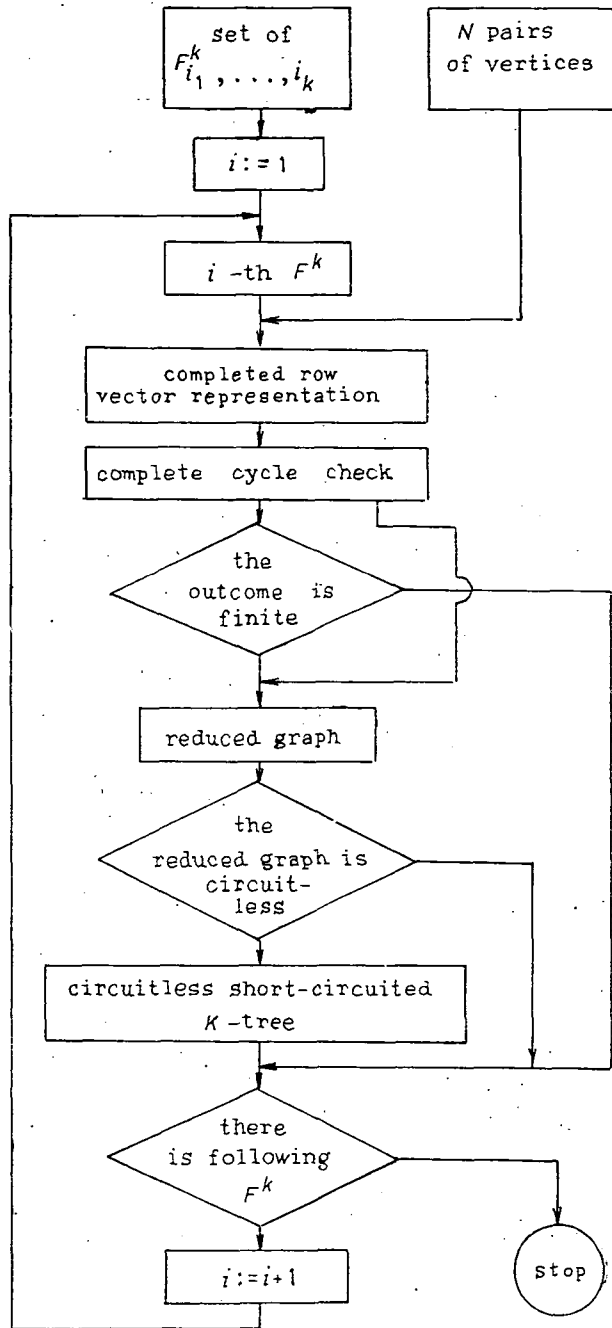


Fig. 1

undirected one, too. Hereupon let us assume that the complete cycle check has a finite outcome and the reduced graph contains a circuit with multiple vertices P_{a_1}, \dots, P_{a_l} . For the reduced graph is bipartite between all adjoining multiple vertices mentioned above there is exactly one selected vertex (P_{a_l} is adjoining P_{a_1} !). Let be P_{i_j} between P_{a_j} and $P_{a_{j+1}}$, where $j=1, \dots, l$ and $l+1=1$. But then a directed path leads either from P_{a_j} to $P_{a_{j+1}}$, or from P_{a_j} and $P_{a_{j+1}}$ to the selected vertex P_{i_j} . Therefore all adjoining multiple vertices are connected in the short-circuited k -tree so it contains a circuit (or a loop). The proof is complete.

In the sequel we are going to construct a procedure that selects short-circuited in N pairs of vertices k -trees which have no circuit (or loop) from among graphs F_{i_1, \dots, i_k}^k :

Step 1. Consider the set of all k -trees F_{i_1, \dots, i_k}^k where i_1, \dots, i_k are the indices of the selected vertices given in advance. From the row vector representation of such k -trees and from the pairs of vertices given by conditions of short-circuiting the complete row vector representation of short-circuited k -trees can be constructed. This means the introduction of symbol elements which are the indices of multiple vertices.

Step 2. A complete cycle check is performed on each representation constructed at the above step. If it has a finite outcome so there exists the reduced graph of the corresponding short-circuited k -tree.

Step 3. In the end we control whether the reduced graph is circuitless. In the circuitless case the corresponding short-circuited k -tree will not have a circuit (or a loop) according to the theorem.

One can look over the whole procedure by studying the block diagram on figure 1. We notice that there is a task in the 3-rd part of the procedure to find out whether a graph is without circuit. It may happen in several different ways. In a simple case it is possible by drawing the reduced graph. Further on it can be found out from the incidence matrix of the graph [1]. As for the reduced graph is bipartite its circuit contains only edges of even numbers and we can search for all possible circuits from the table defining the graph after all. To construct such a discussion can easily be made because it consists of steps of finite numbers.

Application

Example 1. Let be given the row vector representation of a 2-tree generated by method [6]:

$$(202527808).$$

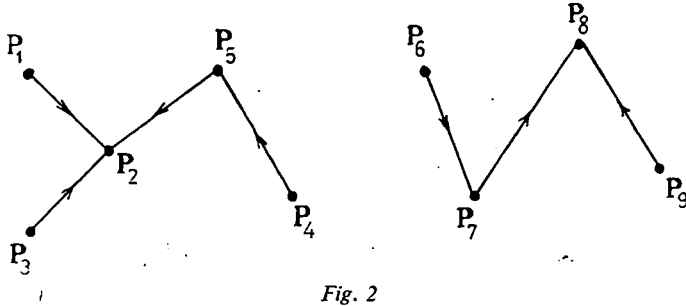
It is obvious from the method [6] that this 2-tree has 9 vertices and its selected vertices are P_2 and P_8 . We can easily draw the directed 2-tree and it is illustrated on the figure 2.

First short-circuit the pairs of vertices P_2, P_6 and P_4, P_8 . Let be defined the symbol elements by equations $2=6=a$ and $4=8=b$. So we get the following completed row vektor representation of the arisen short-circuited 2-tree:

$$\mathbf{R} = \begin{pmatrix} 1 & a & 3 & b & 5 & a & 7 & b & 9 \\ a & 0 & a & 5 & a & 7 & b & 0 & b \end{pmatrix}.$$

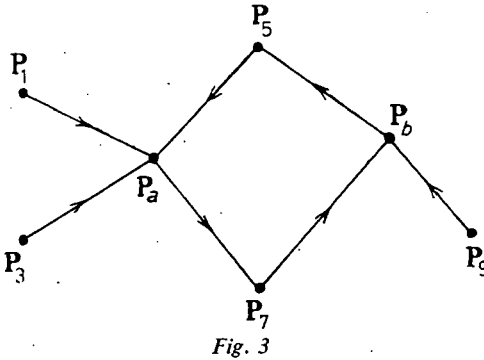
The corresponding short-circuited 2-tree has circuit because of performing a cycle check on R starting at the element "a" being in the 6-th column of R we get the sequence

$$a, 7, b, 5, a, \dots$$



that is the outcome of the cycle check is infinite. Now the reduced graph does not exist according to its definition. The short-circuited 2-tree can be seen in the figure 3. Notice that the cycle check of infinite outcome means "walking along the only directed circuit of the graph".

For the second short-circuit the pairs of vertices P_1, P_4 and P_4, P_7 of the 2-tree. The common symbol element is defined by equation $1=4=7=a$, and the completed row vector representation of the short-circuited 2-tree is:



$$R = \begin{pmatrix} a & 2 & 3 & a & 5 & 6 & a & 8 & 9 \\ 2 & 0 & 2 & 5 & 2 & a & 8 & 0 & 8 \end{pmatrix}$$

The complete cycle check performing on R has finite outcome because of the following possible cycle checks:

$$\begin{aligned} & a, 2, 0, \\ & a, 5, 2, 0, \\ & \text{and } a, 8, 0. \end{aligned}$$

The table of the reduced graph found out from the sequences of cycle checks is the following:

$$\frac{a}{2, 2, 8}$$

The edges of the reduced graph are (P_a, P_2) , (P_a, P_2) and (P_a, P_8) , therefore this graph contains a circuit consisting of two edges. Either the short-circuited graph or its reduced graph are drawn on the figure 4. Notice that there is not a (directed) circuit in the directed graph but it exists in the undirected one. Otherwise this fact turned out from the complete cycle check of finite outcome as well. The present

example shows that the complete cycle check of finite outcome is not a sufficient condition for the short-circuited k -tree to be circuitless.

For the third short-circuit the pairs of vertices P_5, P_6 in the considered 2-tree. Let be defined the symbol element "a" by $5=6=a$. So the representation of the arosen short-circuited 2-tree is:

$$R = \begin{pmatrix} 1 & 2 & 3 & 4 & a & a & 7 & 8 & 9 \\ 2 & 0 & 2 & a & 2 & 7 & 8 & 0 & 8 \end{pmatrix}.$$

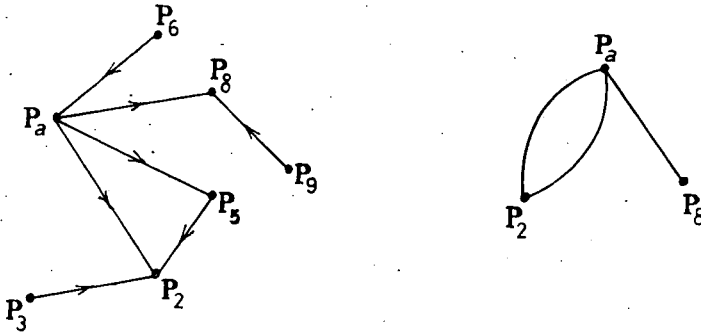


Fig. 4

The steps of the complete cycle check performed on R are:

$$a, 2, 0,$$

$$\text{and } a, 7, 8, 0,$$

so the outcome of the complete cycle check is finite. Because of the reduced graph defined by the table

$$\frac{a}{2, 8}$$

is circuitless so the short-circuited 2-tree is too. We can show either the short-circuited 2-tree or its reduced graph on the figure 5.

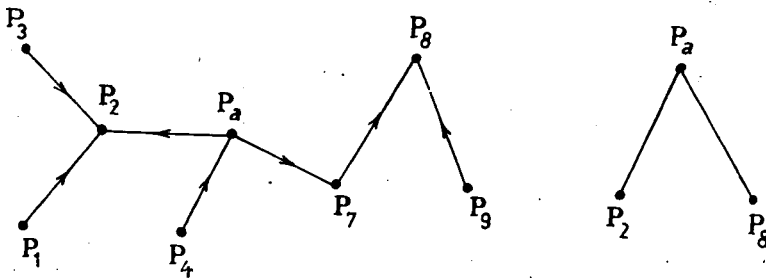


Fig. 5

Example 2. Consider the graph with 6 vertices marked by natural numbers on the figure 6. This is a network graph of a feedback operational amplifier with multiple loops which plays central part in the theory of the linear active electrical networks [2]. The network graph in question is a so called "nullator-norator equiv-

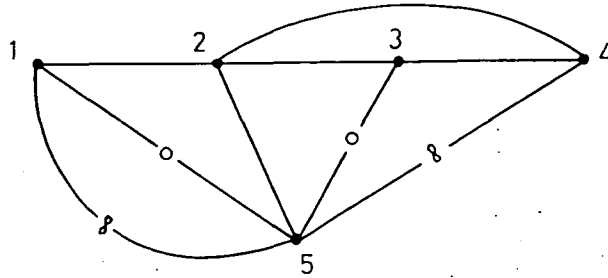


Fig. 6

alent network" [3]. The edges drawn in usual way mean the present of passive element while the symbol "—○—" on the edges refers to a nullator, and the "— 8 —" to a norator. Search all the common 3-trees that become connected circuitless subgraphs after short-circuiting either all the norators or all the nullators while passing over at first all nullators for the second all norators from the graph of network.

First we produce those trees of the graph which contain all the norators and do not contain any of the nullators. Such trees of the graph can be easily produced by method [6]. If we pass over all the norators from the subgraph mentioned above we really get all the 3-trees of the graph which become connected circuitless subgraphs after short-circuiting of endpoints of the norators. After passing over all the norators we get the following result:

(01200), (01400), (03400), (04200), (04400), (05200), (05400).

In the 2-nd step short-circuit the endpoints of all nullators in each of the 3-trees listed above. This means the short-circuiting of pairs of vertices 1,5 and 3,5. From the short-circuited 3-trees arisen by fusion of vertices 1,3 and 5 the circuitless graphs can be selected by method shown in the present paper.

So the only symbol element "a" is defined by equation $1=5=3=a$. The 7-completed row vector representation can be written as following: the first common row of the representation are

a 2 a 4 a

while the second rows are in order

0 a 2 0 0
 0 a 4 0 0
 0 a 4 0 0
 0 4 2 0 0
 0 4 4 0 0
 0 a 2 0 0

and 0 a 4 0 0.

Complete cycle check performed on each of representation in question we get an infinite outcome only in the 1-st and in the 6-th cases. In the remaining cases the reduced graph is a common one defined by table

$$\frac{a}{1, 4, 5}$$

and it is obviously circuitless.

We obtain the following 3-trees fulfilling the conditions of the present example: (01400), (03400), (04200), (04400), (05400).

References

RESEARCH GROUP ON MATHEMATICAL LOGIC
AND THEORY OF AUTOMATA OF THE
HUNGARIAN ACADEMY OF SCIENCES
H-6720 SZEGED, HUNGARY
SOMOGYI U. 7.

References

- [1] BUSACKER, R. G. & T. L. SAATY, *Finite graphs and networks*, McGraw—Hill Book Comp., New York, 1967.
- [2] CHIRLIAN, P., *Integrated and active network analysis*, McGraw—Hill Book Comp., New York, 1969.
- [3] DAVIES, A. C., Norator-nullator equivalent networks for controlled sources, *Proc. IEEE*, v. 55, 1967, p. 772.
- [4] DAVIES, A. C., Matrix analysis of network containing nullors, *Electronics Letters*, 1966, No. 2, p. 48, No. 3. p. 91.
- [5] ORE, O., *Theory of graphs*, Amer. Math. Soc., Transl. Providence, 1962, pp. 68—74.
- [6] PÁVÓ, I., Generation of the k -trees of a graph, *Acta Cybernet.*, v. 1, 1971, pp. 57—68.
- [7] TELLEGEN, B. D. H., On nullators and norators, *IEEE Trans. Circuit Theory*, v. CT—13, 1966, pp. 466.

(Received Sept. 5, 1974)