The research reported in this document was made possible by support extended to the University of Illinois, Coordinated Science Laboratory, jointly by the Department of the Army, Department of the Navy (Office of Naval Research), and the Department of the Air Force (Office of Scientific Research) under Department of Army Contract DA-28-043-AMC-00073(E).
Abstract

GENERATION OF DIRECTED TREES, 2-TREES AND PATHS WITHOUT DUPLICATION

Archie Joseph Paul, Jr., Ph.D.
Department of Electrical Engineering
University of Illinois, 1965

The increasing number of applications of graph theory to the solution of problems in many fields make it desirable to have available complete knowledge of the properties of these graphs. Since many problems in electrical networks, switching circuits, and communication nets can be formulated in terms of directed graphs, it is appropriate to study their properties.

In this paper, procedures are developed for generating the directed trees, 2-trees and paths of a directed graph. Unlike other methods for generating these subgraphs, the procedures developed here avoid generating duplicate elements thus they eliminate the necessity of repeated search to select a complete set of elements. Proofs are given to verify that all elements of the set of directed trees, 2-trees or paths are generated and that no duplicate elements occur. Examples are given to illustrate the procedures in detail.

The procedures are amenable to digital computer application.
The author wishes to express his gratitude to his advisor, Professor W. Mayeda, for his invaluable advice and encouragement throughout the period of research for this thesis.

He is also grateful for the support rendered by the Coordinated Science Laboratory of the University of Illinois.
TABLE OF CONTENTS

Chapter |
-------|
I. INTRODUCTION | 1 |
II. GENERATION OF DIRECTED TREES WITHOUT DUPLICATION | 2 |
   2.1 Introduction | 2 |
   2.2 A Procedure for Generating the Directed Trees of G | 3 |
   2.3 Example 1 | 6 |
   2.4 The Order of the Edges of the Reference Tree | 8 |
   2.5 Verification that all Trees can be Generated | 10 |
   2.6 Generation of Distinct Trees | 12 |
   2.7 Example 2 | 13 |
III. GENERATION OF DIRECTED SIMPLE PATHS WITHOUT DUPLICATION | 19 |
   3.1 Introduction | 19 |
   3.2 Obtaining the Path from a Tree | 21 |
   3.3 Generation of a Minimal Set of Trees Containing $P_{ij}$ | 22 |
   3.4 Modification of a Tree to Maximize $S_{ai}(t) \cap \xi(a_i)$ for a Given Path | 26 |
   3.5 Verification that all Paths are Generated | 29 |
   3.6 No Duplicate Paths are Generated | 30 |
   3.7 Example | 31 |
IV. GENERATION OF DIRECTED 2-TREES WITHOUT DUPLICATION | 36 |
   4.1 Introduction | 36 |
   4.2 Generation of the Set of 2-trees, $T_{2d_{k:j}}$ | 36 |
   4.3 Generation of the Set of 2-trees, $T_{2d_{1,k:j}}$ | 38 |
   4.4 Example | 40 |
V. CONCLUDING REMARKS | 42 |
BIBLIOGRAPHY | 43 |
VITA | 44 |
CHAPTER I

INTRODUCTION

Graph theory as a mathematical discipline finds broad application in the solution of problems of many fields. It is of particular interest to Electrical Engineers for its applicability to the solution of problems in electrical networks, communication nets, and switching circuits. A large number of these problems can be formulated in terms of directed graphs [1,2,6,7]; consequently the properties of such graphs is of major importance. This paper discusses three properties of directed graphs, namely, directed trees, 2-trees and paths.

The problem of the generation of trees without duplication for the non oriented graph has recently been solved [8]. The concept of ordering the edges of the reference tree which was developed in the solution of that problem is also employed in the solution of the analogous problem for the directed graph, although many of the conditions imposed are different due to the nature of the directed tree.

Only connected directed graphs having a finite number of vertices and edges are considered. The definitions and symbols used are for the most part those used in the literature.
CHAPTER II
GENERATION OF DIRECTED TREES WITHOUT DUPLICATION

2.1 Introduction

A directed tree is defined as a tree which has one vertex specified as a base vertex and which contains a simple directed path from every vertex to the base vertex. It can be seen that the directed trees of a graph, G, depend upon the base vertex specified.

It is convenient to define the following sets in G to clarify the notation used.

\[ \Omega(G) = \{ w : w \text{ is a vertex of } G \} \] (2.1)
\[ D(j) = \{ x : x \in \Omega(G) \text{ and } \exists \text{ a directed path from } j \text{ to } x \} \] (2.2)
\[ D^*(j) = \{ y : y \in \Omega(G) \text{ and } j \in D(y) \} \] (2.3)

The subset \( D(j) \) is referred to as the set of vertices accessible from \( j \) and \( D^*(j) \) is the set of vertices from which \( j \) is accessible.

\[ E(G) = \{ e : e \text{ is an edge of } G. \} \] (2.4)
\[ S_e(t) = \{ x : x \text{ is an edge in the fundamental cut-set of } G \text{ with respect to the tree, } t, \text{ and containing edge } e \text{ of } t. \} \] (2.5)
\[ \xi(x) = \{ y : y \text{ is an edge of } G \text{ having the same initial vertex as edge } x. \} \] (2.6)

The symbol \( \oplus \) is used to indicate the symmetric difference of sets, that is,

\[ A \oplus B = (A-B) \cup (B-A) \text{ where } A-B = \{ a : a \in A \text{ and } a \notin B \}. \] (2.7)
2.2 A Procedure for Generating the Directed Trees of G

A necessary and sufficient condition for the existence of a directed tree of G having vertex j as base vertex is \( D^*(j) = \Omega(G) \). When this condition is satisfied, one can choose a directed tree,

\[
    t_o = \{ e_1, e_2, e_3, \ldots, e_{v-1} \},
\]

as a reference tree of the graph, G, having v vertices and use the following Theorem to generate a set of directed trees having the same base vertex.

**Theorem I:** A distinct directed tree, \( t \), of G is formed from \( t_o \) by replacing edge \( e_i \in t_o \) by an edge \( e \in E(G) \) iff \( e \in S_{e_i}(t_o) \cap \xi(e_i) \oplus \{ e_i \} \). A set of distinct trees, \( T_e \), of G is formed by repeating this replacement for each element of \( S_{e_i}(t_o) \cap \xi(e_i) \oplus \{ e_i \} \).

**Proof:** \( \implies \) Assume \( e \in S_{e_i}(t_o) \cap \xi(e_i) \oplus \{ e_i \} \). We need only show that the subgraph, \( t \), resulting from the replacement is a directed tree; it is obvious that it is distinct because \( e_i \in t_o \) and \( e_i \notin t \).

The removal of \( e_i \) from \( t_o \) separates \( t_o \) into two subgraphs, A and B. If the initial vertex of \( e_i \) is in A, it is accessible from all vertices in A while vertex j is in B and is accessible from all vertices in B. By definition of "fundamental cut-set" all elements in \( S_{e_i}(t_o) \) either connect one vertex in A to a vertex in B or vica versa. Hence any element in \( S_{e_i}(t_o) \cap \xi(e_i) \oplus \{ e_i \} \) could replace \( e_i \) to form a new tree, but it would not necessarily be a directed tree; however, if \( e \in S_{e_i}(t_o) \cap \xi(e_i) \oplus \{ e_i \} \) it connects the initial vertex of \( e_i \) in A to some vertex in B and in the
tree formed there exists a directed path from each vertex to the base vertex \( j \). Hence it is a directed tree.

\[ \iff \text{ Assume that } e_i \in \mathcal{T}_o \text{ and } e_i \text{ is replaced by } e \text{ where } e \in G - \mathcal{T}_o \text{ and the resulting subgraph is a directed tree. It must be shown that } e \in S_{e_i}^e (\mathcal{T}_o) \cap \xi(e_i) \oplus \{e_i\}. \]

In order for \( \mathcal{T} \) to be a directed tree of \( G \) the element, \( e \), replacing \( e_i \) must be a member of \( \xi(e_i) \). This follows from the knowledge that in every directed tree there exists exactly one edge issuing from each vertex of \( G \) other than the base vertex \( j \). Hence it is necessary that \( e_i \) be replaced by an element in \( \xi(e_i) \) otherwise the resulting subgraph, \( \mathcal{T} \), would have a vertex, the initial vertex of \( e_i \), other than the base vertex, \( j \), which would not be the initial vertex of any edge belonging to \( \mathcal{T} \); therefore no directed path could exist from that vertex to the base vertex and the subgraph would not be a directed tree.

It is also necessary that \( e \in S_{e_i}^e (\mathcal{T}_o) \) for the following reason. Removal of \( e_i \) from \( \mathcal{T}_o \) separates the subgraph, \( \mathcal{T}_o \), into two parts. The edge, \( e \), replacing \( e_i \) must connect these two parts if the resulting subgraph is to be a tree. Since all edges of \( G \) connecting these two parts belong to \( S_{e_i}^e (\mathcal{T}_o) \), we conclude that \( e \in S_{e_i}^e (\mathcal{T}_o) \cap \xi(e_i) \oplus \{e_i\} \). Q.E.D.

The application of Theorem I to \( \mathcal{T}_o \) will generate the set of distinct trees

\[ T_{e_i}^e = \{ t : t = \mathcal{T}_o \oplus \{e_i, e\}, e_i \in \mathcal{T}_o, e \in S_{e_i}^e (\mathcal{T}_o) \cap \xi(e_i) \oplus \{e_i\} \}. \quad (2.9) \]
Additional distinct directed trees can be formed by replacing another edge of \( t_0 \) in each tree, \( t \), in the set \( T^i \). In this case we replace edge \( e_i' \) which belongs to \( t \cap t_0 \) by the elements of \( S_{e_i'}(t) \cap \xi(e_i') \) one at a time forming as many new trees as there are elements in \( \{e_i'\} \oplus S_{e_i'}(t) \cap \xi(e_i') \). This set of distinct directed trees differing from \( t_0 \) by two edges we label \( T_{i1}^{e_i e_i'} \) as both edges \( e_i \) and \( e_i' \) of \( t_0 \) have been replaced.

\[
T_{i1}^{e_i e_i'} = \{ t' : t' = t \oplus \{e_i', e_i' \}, e_i' \in t_0 \cap t, t \in T^i, e_i' \in S_{e_i'}(t) \cap \xi(e_i') \oplus \{e_i'\} \}
\]  

(2.10)

The order of \( e_i, e_i' \) is such that duplication is avoided. Ordering of the edges of \( t_0 \) is discussed in section 2.4.

This procedure can be continued to obtain all distinct trees differing from \( t_0 \) by three edges, four edges, etc. up to \( v-1 \) edges. The complete set of all distinct directed trees is

\[
T_j = \{t_0, T^1, T_{12}, T_{123}, \ldots, T_{123 \ldots v-1} \}
\]  

(2.11)

where the general term is

\[
T_{i12 \ldots i_k}^{e_i e_i' \ldots e_i k} = \{ t : t = t' \oplus \{e_i', e_i' \}, e_i' \in t' \cap t_0, e \in S_{e_i'}(t') \cap \xi(e_i') \oplus \{e_i'\}, e_i' \in S_{e_i'}(t') \cap \xi(e_i') \oplus \{e_i'\}, t' \in T_{i12 \ldots i_{k-1}} \}
\]  

(2.12)
The sequence, \( \{e_{i_1}, e_{i_2}, \ldots, e_{i_k}\} \), is a subset of the \( v-1 \) edges of \( t_o \) ordered in the manner described in section 2.4.

Before demonstrating that this procedure generates all distinct directed trees of \( G \) and that no duplicate trees are generated, a simple example is given to illustrate the procedure.

2.3 Example 1

Assume vertex 2 is the base vertex and find all directed trees of the graph, \( G \), in Figure 1.

To find the directed trees let \( t_o = \{abc\} \) and compute

![Diagram of a directed graph with vertices labeled 1, 2, 3, 4, 5, 6, 7 and edges labeled a, b, c, d, e, f, g.]

Figure 1. A Directed Graph, \( G \).

the classes of trees, \( T_{i_1}, T_{i_1i_2}, T_{i_1i_2i_3}, \) and \( T_{i_1i_2i_3i_4} \)

using equation 2.12. We proceed by computing the following sets.
From these sets

$$e_1^1 t = \{t^a, t^e\} = \{fbec, abcg\} = \{t_1, t_2\}.$$  

For the directed trees differing from $t_o$ by two edges we compute the following sets.

$$S_b(t_1) \cap \xi(b) \oplus \{b\} = \{a, b, g\} \cap \{b\} \oplus \{b\} = \emptyset$$

$$S_c(t_1) \cap \xi(c) \oplus \{c\} = \{c, d\} \cap \{c\} \oplus \{c\} = \emptyset$$

$$S_e(t_1) \cap \xi(e) \oplus \{e\} = \{a, d, f, g\} \cap \{e, g\} \oplus \{e\} = \{g\}$$

$$S_c(t_2) \cap \xi(c) \oplus \{c\} = \{c, d\} \cap \{c\} \oplus \{c\} = \emptyset$$

From these sets

$$e_1^1 e_2^1 = \{t^a e\} = \{fbgc\} = \{t_3\}.$$

For the directed trees differing from $t_o$ by three branches we compute the following sets.

$$S_c(t_3) \cap \xi(c) \oplus \{c\} = \{c, d\} \cap \{c\} \oplus \{c\} = \emptyset$$

From this set

$$e_1^1 e_2^1 e_3^1 = \emptyset$$

and it follows that

$$e_1^1 e_2^1 e_3^1 e_4^1 = \emptyset.$$  

The set of all directed trees with base vertex 2 is now obtained from equation 2.11.

$$T_2 = \{t_o, t^a, t^e, t^{ae}\} = \{t_0, t_1, t_2, t_3\} = \{abec, fbec, abcg, fbgc\}$$
2.4 The Order of the Edges of the Reference Tree

The sets of trees differing from $t^o_0$ by two edges in Example 1 were $T^{ab}, T^{ae}, T^{ac}$, and $T^{ec}$. Since the sets $T^{ab}, T^{ac}$ and $T^{ec}$ were empty, we were left with the set $T^{ae}$. The sets $T^{ea}$ and $T^{eb}$ were not computed as they would contribute only duplicate trees. In this example $T^{ea} = T^{ae}$. This duplication was avoided by ordering the elements of $t^o_0$, that is, $t^o_0 = \{abc\}$, where $a < b < e < c$, and then in the sets of trees, $T^{e1}e2...e_k$, permitting only the sequences, $\{e_{i1}, e_{i2}, ..., e_{ik}\}$, which satisfy $e_{i1} < e_{i2} < ... < e_{ik}$, where $1 \leq k \leq v-1$.

This order can be achieved by ordering the edges of $t^o_0$ so that an edge sequence from any vertex to the base vertex is in increasing order. There are numerous ways of ordering $t^o_0$ to satisfy this requirement; one method is as follows. Partition the edges of $t^o_0$ into disjoint subsets, $A_i$, such that

\begin{align*}
A_1 &= \{e : e \in t^o_0 \cap \bar{\phi}(j)\} \\
A_i &= \{e : e \in t^o_0 \cap \delta(A_{i-1})\}, \text{ for } i = 2, 3, ..., n, \tag{2.13}
\end{align*}

where $n$ is the length of the longest path in $t^o_0$,

\begin{align*}
\bar{\phi}(j) &= \{e : e \in G \text{ and the terminal vertex of } e \text{ is } j\}\text{ and} \\
\delta(A) &= \{e : e \in G \text{ and } e \text{ has the same terminal vertex as the initial vertex of some edge, } x, \text{ contained in } A\}. \tag{2.14}
\end{align*}

The edges are then ordered by first assigning order to the elements of $A_n$ in arbitrary fashion starting with the lowest order and increasing for
each element of $A_n$, then proceed to $A_{n-1}$ assigning increasing order to
the elements of $A_{n-1}$ in arbitrary fashion. Continue this procedure for
$A_{n-2}$, $A_{n-3}$, etc. until $t_o$ is ordered. Figure 2 illustrates this
procedure.

![Figure 2. An Ordered Directed Tree.](image)

$A_1 = \phi(j) \cap t_o = \{v,w\}$

$A_2 = \delta(A_1) \cap t_o = \{z,y,m,s\}$

$A_3 = \delta(A_2) \cap t_o = \{q,r,n,k\}$

$A_4 = \delta(A_3) \cap t_o = \{u,p,f,h\}$

$A_5 = \delta(A_4) \cap t_o = \{x,c,d,g\}$

$A_6 = \delta(A_5) \cap t_o = \{a,b\}$

One possible ordering of $t_o$ is

$$t_o = \{a,b,x,c,d,g,u,p,f,h,q,r,n,k,z,y,m,s,v,w\}.$$  (2.15)

It will now be shown that if $t_o = \{e_1,e_2,e_3,\ldots,e_{v-1}\}$, where
e_1 < e_2 < e_3 < \ldots < e_{v-1}, and in the sets of trees,
The sequence, \( \{i_1, i_2, i_3, \ldots, i_k\} \), is a subset of the natural numbers \( \{1, 2, 3, \ldots, v-1\} \) such that \( i_1 < i_2 < i_3 < \ldots < i_k \), then all distinct trees are generated without duplication by the procedure given in section 2.2.

2.5 Verification that all Trees can be Generated

**Lemma I:** Given an ordered reference directed tree, \( t_0 \), of a directed graph, \( G \), then for any arbitrary directed tree, \( t \), differing from \( t_0 \) by \( k \) branches there exists a directed tree, \( t' \), differing from \( t_0 \) by \( k-1 \) branches from which \( t \) can be obtained from the equation

\[
t = t' \oplus \{e_i, e\}
\]

where \( e_i \in t' \cap t_0 \), \( e \in S_{e_i} (t') \cap \xi(e_i) \oplus \{e_i\} \) and \( e_i \) is of higher order than any element of \( t_0 \) which is not contained in \( t' \).

**Proof:** Consider the arbitrary tree, \( t \), and the reference tree, \( t_0 \), with base vertex, \( j \). Since \( t \neq t_0 \), we can start at vertex \( j \) and examine the edges of \( t_0 \) in reverse order starting with the highest order edge and proceeding in decreasing order of the edges of \( t_0 \) until we find an edge, \( e_i \), such that \( e_i \in t_0 - t \). All edges of \( t_0 \) in the directed edge sequence from the terminal vertex of \( e_i \) to vertex \( j \) are of higher order and they belong to \( t \) as \( e_i \) is the edge of highest order which belongs to \( t_0 \) but does not belong to \( t \). This is illustrated in Figure 3.

![Figure 3. Tree, t, and Reference Tree, t_o](image-url)
Since $e_i \notin t$, there exists an edge $\alpha_k \ni \alpha_k \in \xi(e_i) \cap t$. This follows from the fact that every directed tree has one edge issuing from each vertex other than the base vertex. Removal of $\alpha_k$ from $t$ separates the tree, $t$, into two parts. The terminal vertex of branch $e_i$ belongs to the subset of vertices from which $j$ is accessible in $t$ after the removal of $\alpha_k$; the same is true of the terminal vertex of $\alpha_k$. The initial vertex of edges $e_i$ and $\alpha_k$ is accessible from all vertices in the part of $t$ not containing $j$. Hence $e_i \in \xi(\alpha_k)$ and $e_i \in S_{\alpha_k}(t)$; consequently there exists a tree $t'$ formed from $t$ by replacing $\alpha_k$ with $e_i$. This tree has one more element in common with $t_o$, and $t$ can be generated from $t'$ by equation 2.16 if $\alpha_k = e$. The conditions necessary, $\alpha_k \in S_{e_i}(t') \cap \xi(e_i) \oplus [e_i]$, $e_i \in t' \cap t_o$ and $e_i$ is of higher order than any element of $t_o$ not contained in $t'$, are all satisfied. Q.E.D.

**Theorem II:** All directed trees of $G$ are generated by the procedure given in section 2.2.

**Proof:** It is clear that every tree differing from $t_o$ by one edge can be obtained from $t_o$ by replacing an edge of $t_o$ as per equation 2.12. If it is assumed that an arbitrary tree, $t'$, differing from $t_o$ by $k-1$ branches, ($2 < k < v$), can be obtained by equation 2.12, then by Lemma I it can be shown that an arbitrary tree, $t$, differing from $t_o$ by $k$ branches can be obtained by equation 2.12; hence by the principle of weak induction we prove that all directed trees can be generated by the procedure given in section 2.2. Q.E.D.
2.6 Generation of Distinct Trees

It can be seen that in generating the classes of directed trees
\[ e_{11} e_{12} \ldots e_{i_k} \]
there can be no tree appearing in more than one of the classes
\[ e_{11}, e_{12}, \ldots, e_{i_k} \]
This can be seen if you consider that in each class of trees the number of edges of each tree common to \( t_o \) is specified. In replacing edges of \( t_o \) to generate the trees the edge replacing an edge of \( t_o \) does not belong to \( t_o \) since it must initiate from the same vertex as the edge it replaces and must be distinct from it. Hence the trees of each class have a different number of edges in common with \( t_o \); consequently no tree can be common to more than one class and of course no tree generated can be equal to \( t_o \).

Within a class how do we know that there are no duplicate trees?

To answer this question consider the sets of trees in the class
\[ e_{11} e_{12} \ldots e_{i_k} \]
It is clear that if
\[ e_{11}, e_{12}, \ldots, e_{i_k} \neq e_{j_1}, e_{j_2}, \ldots, e_{j_k} \] (2.17)
then no tree can be common to the two sets and
\[ e_{11} e_{12} \ldots e_{i_k} \cap e_{j_1} e_{j_2} \ldots e_{j_k} = \emptyset \] (2.18)
because all trees in the first set do not have edges \( e_{11}, e_{12}, \ldots, e_{i_k} \)
while all trees in the second set do not contain edges \( e_{j_1}, e_{j_2}, \ldots, e_{j_k} \).

If two trees are duplicates they contain the same branches and therefore
any branch of G missing from one tree must be missing from the other.

It follows then that if there are duplicate trees they must occur
within a class $T_{i_1 i_2 \ldots i_k}$. Due to the ordering requirement imposed
on the set \{e_{i_1}, e_{i_2}, \ldots, e_{i_k}\} there is only one class of trees having
this set of edges missing, and it is generated by replacing the edges
of $t_0$ in the order given. Duplicate trees would have to result from
replacing $e_{i_k}$ with the edges in the set \{e : e \in S_{i_k} \cap (e_{i_k} \oplus \{e_{i_k}\})\}.

This is not possible as the replacement procedure generates a set of
trees each having a distinct edge issuing from the initial vertex, $x$,
of $e_{i_k}$. Since each directed tree has only one edge issuing from
vertex $x$ and all trees in the set have a different edge issuing from
this vertex there can not be duplicates within a class of directed trees.

2.7 Example 2

The following example illustrates the procedure. Given the
directed graph G, of Figure 4, find all directed trees having vertex 7
as base vertex.

Figure 4, A Directed Graph G.
Let the reference tree \( t_o = \{abcdef\} \). The necessary ordering of \( t_o \) is achieved by forming the following partitions.

\[
A_1 = \delta(7) \cap t_o = \{e,f\} \cap \{a,b,c,d,e,f\} = \{e,f\}
\]

\[
A_2 = \delta(e) \cup \delta(f) \cap t_o = \{c,k\} \cup \{o,g\} \cap \{a,b,c,d,e,f\} = \{c\}
\]

\[
A_3 = \delta(c) \cap t_o = \{d,n,b,p\} \cap \{a,b,c,d,e,f\} = \{b,d\}
\]

\[
A_4 = \delta(b) \cup \delta(d) \cap t_o = \{a\} \cup \{h,i,j,l\} \cap \{a,b,c,d,e,f\} = \{a\}
\]

\[
A_5 = \delta(a) \cap t_o = \{m\} \cap \{a,b,c,d,e,f\} = \{\emptyset\}
\]

The order is now assigned as discussed in section 2.4 to give \( t_o \) in proper order as \( t_o = \{abdcef\} \).

The following sets are calculated to determine \( T_{e_1l} \).

\[
S_a(t_o) \cap \xi(a) \oplus \{a\} = \{a,n,k,m\} \cap \{a,n,k\} \oplus \{a\} = \{n,k\}
\]

\[
S_b(t_o) \cap \xi(b) \oplus \{b\} = \{m,l,k,n,b\} \cap \{b,l\} \oplus \{b\} = \{l\}
\]

\[
S_d(t_o) \cap \xi(d) \oplus \{d\} = \{m,l,d,j,i,h\} \cap \{m,d\} \oplus \{d\} = \{m\}
\]

\[
S_c(t_o) \cap \xi(c) \oplus \{c\} = \{c,p,k,j,i,h\} \cap \{c\} \oplus \{c\} = \{\emptyset\}
\]

\[
S_e(t_o) \cap \xi(e) \oplus \{e\} = \{e,h,i,o\} \cap \{o,j,p,e\} \oplus \{e\} = \{o\}
\]

\[
S_f(t_o) \cap \xi(f) \oplus \{f\} = \{f,g,h,o\} \cap \{f,h\} \oplus \{f\} = \{h\}
\]

The trees differing from \( t_o \) by one branch are

\[
T_{1}^{e_1l} = \{T^a, T^b, T^d, T^e, T^f\}
\]

where

\[
T^a = \{t_o \oplus \{a,n\}, t_o \oplus \{a,k\}\} = \{nbdcf, kbdcef\} = \{t_1, t_2\},
\]

\[
T^b = \{t_o \oplus \{b,l\}\} = \{agdcef\} = \{t_3\},
\]

\[
T^d = \{t_o \oplus \{d,m\}\} = \{abmcef\} = \{t_4\},
\]

\[
T^e = \{t_o \oplus \{e,o\}\} = \{abdcof\} = \{t_5\},
\]

\[
T^f = \{t_o \oplus \{f,h\}\} = \{abdceh\} = \{t_6\}.
\]
The following sets are now calculated to evaluate $T^{e_1 e_2}$.

\[ \begin{align*}
S_b(t_1) \cap \xi(b) \oplus \{b\} &= \{a, b, e \} \cap \{b, \ell \} \oplus \{b\} = \{\ell\}, \\
S_d(t_1) \cap \xi(d) \oplus \{d\} &= \{d, m, j, i, h \} \cap \{d, m\} \oplus \{d\} = \{m\}, \\
S_e(t_1) \cap \xi(e) \oplus \{e\} &= \{e, i, o, h \} \cap \{o, j, p, e\} \oplus \{e\} = \{o\}, \\
S_f(t_1) \cap \xi(f) \oplus \{f\} &= \{h, o, f, g \} \cap \{f, h\} \oplus \{f\} = \{h\}, \\
S_b(t_2) \cap \xi(b) \oplus \{b\} &= \{a, b, \ell \} \cap \{b, \ell \} \oplus \{b\} = \{\ell\}, \\
S_d(t_2) \cap \xi(d) \oplus \{d\} &= \{d, m, j, i, h \} \cap \{d, m\} \oplus \{d\} = \{m\}, \\
S_e(t_2) \cap \xi(e) \oplus \{e\} &= \{e, i, o, h \} \cap \{o, j, p, e\} \oplus \{e\} = \{o\}, \\
S_f(t_2) \cap \xi(f) \oplus \{f\} &= \{h, o, f, g \} \cap \{f, h\} \oplus \{f\} = \{h\}, \\
S_d(t_3) \cap \xi(d) \oplus \{d\} &= \{h, i, j, k, d, n, b \} \cap \{m, d\} \oplus \{d\} = \{\emptyset\}, \\
S_e(t_3) \cap \xi(e) \oplus \{e\} &= \{h, i, o, e \} \cap \{o, j, p, e\} \oplus \{e\} = \{o\}, \\
S_f(t_3) \cap \xi(f) \oplus \{f\} &= \{h, o, f, g \} \cap \{h, f\} \oplus \{f\} = \{h\}, \\
S_e(t_4) \cap \xi(e) \oplus \{e\} &= \{h, i, o, e \} \cap \{o, j, p, e\} \oplus \{e\} = \{o\}, \\
S_f(t_4) \cap \xi(f) \oplus \{f\} &= \{h, o, f, g \} \cap \{h, f\} \oplus \{f\} = \{h\}, \\
S_f(t_5) \cap \xi(f) \oplus \{f\} &= \{f, g, i, e \} \cap \{h, f\} \oplus \{f\} = \{\emptyset\}.
\end{align*} \]

$T^{e_1 e_2} = \{T_{ab}, T_{ad}, T_{ae}, T_{af}, T_{be}, T_{bf}, T_{de}, T_{df}\}$ where

\[ \begin{align*}
T_{ab} &= \{t_1 \oplus \{b, \ell\}, t_2 \oplus \{b, \ell\}\} = \{nldcef, kldcef\} = \{t_7, t_8\}, \\
T_{ad} &= \{t_1 \oplus \{d, m\}, t_2 \oplus \{d, m\}\} = \{nbmcef, kbmcef\} = \{t_9, t_{10}\}, \\
T_{ae} &= \{t_1 \oplus \{e, o\}, t_2 \oplus \{e, o\}\} = \{nbdcof, kbdcof\} = \{t_{11}, t_{12}\}, \\
T_{af} &= \{t_1 \oplus \{f, h\}, t_2 \oplus \{f, h\}\} = \{nbdcgeh, kbdceh\} = \{t_{13}, t_{14}\}, \\
T_{be} &= \{t_3 \oplus \{e, o\}\} = \{aldcof\} = \{t_{15}\}, \\
T_{bf} &= \{t_3 \oplus \{f, h\}\} = \{aldcgeh\} = \{t_{16}\}, \\
T_{de} &= \{t_4 \oplus \{e, o\}\} = \{abmcof\} = \{t_{17}\}, \\
T_{df} &= \{t_4 \oplus \{f, h\}\} = \{abmceh\} = \{t_{18}\}.
\end{align*} \]
To evaluate $T$ we calculate the following sets.

$S_d(t_7) \cap \xi(d) \oplus \{d\} = \{a,b,m,d,h,i,j\} \cap \{m,d\} \oplus \{d\} = \{m\}$

$S_e(t_7) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{e,o,j,p\} \oplus \{e\} = \{o\}$

$S_f(t_7) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_d(t_8) \cap \xi(d) \oplus \{d\} = \{a,b,m,d,h,i,j\} \cap \{m,d\} \oplus \{d\} = \{m\}$

$S_e(t_8) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{o,e,j,p\} \oplus \{e\} = \{o\}$

$S_f(t_8) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_e(t_9) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{o,e,j,p\} \oplus \{e\} = \{o\}$

$S_f(t_9) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_e(t_{10}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{o,e,j,p\} \oplus \{e\} = \{o\}$

$S_f(t_{10}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_e(t_{11}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_f(t_{11}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_e(t_{12}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_f(t_{12}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_e(t_{13}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_f(t_{13}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_e(t_{14}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_f(t_{14}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_e(t_{15}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_f(t_{15}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_e(t_{16}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_f(t_{16}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_e(t_{17}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$S_f(t_{17}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\}$

$$e_{i_1}e_{i_2}e_{i_3} = \{T_{abd}^{T}, T_{abe}^{T}, T_{abf}^{T}, T_{ade}^{T}, T_{adf}^{T}\}$$

$T_{abd}^{T} = \{t_7 \oplus \{d,m\}, t_8 \oplus \{d,m\}\} = \{n\&mcef, k\&mcef\} = \{t_{19}, t_{20}\}$

$T_{abe}^{T} = \{t_7 \oplus \{e,o\}, t_8 \oplus \{e,o\}\} = \{n\&dcof, k\&dcof\} = \{t_{21}, t_{22}\}$

$T_{abf}^{T} = \{t_7 \oplus \{f,h\}, t_8 \oplus \{f,h\}\} = \{n\&dceh, k\&dceh\} = \{t_{23}, t_{24}\}$

$T_{ade}^{T} = \{t_9 \oplus \{e,o\}, t_{10} \oplus \{e,o\}\} = \{n\&mcof, k\&mcof\} = \{t_{25}, t_{26}\}$

$T_{adf}^{T} = \{t_9 \oplus \{f,h\}, t_{10} \oplus \{f,h\}\} = \{n\&mceh, k\&mceh\} = \{t_{27}, t_{28}\}$

We now calculate the following sets to evaluate $T$.
\[ S_e(t_{19}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{o,e,j,p\} \oplus \{e\} = \{o\} \]
\[ S_f(t_{19}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\} \]
\[ S_e(t_{20}) \cap \xi(e) \oplus \{e\} = \{h,i,o,e\} \cap \{o,e,j,p\} \oplus \{e\} = \{o\} \]
\[ S_f(t_{20}) \cap \xi(f) \oplus \{f\} = \{h,o,f,g\} \cap \{h,f\} \oplus \{f\} = \{h\} \]
\[ S_f(t_{21}) \cap \xi(f) \oplus \{f\} = \{f,g,i,e\} \cap \{h,f\} \oplus \{f\} = \{\emptyset\} \]
\[ S_f(t_{22}) \cap \xi(f) \oplus \{f\} = \{f,g,i,e\} \cap \{h,f\} \oplus \{f\} = \{\emptyset\} \]
\[ S_f(t_{25}) \cap \xi(f) \oplus \{f\} = \{f,g,i,e\} \cap \{h,f\} \oplus \{f\} = \{\emptyset\} \]
\[ S_f(t_{26}) \cap \xi(f) \oplus \{f\} = \{f,g,i,e\} \cap \{h,f\} \oplus \{f\} = \{\emptyset\} \]
\[ e_{i_1} e_{i_2} e_{i_3} e_{i_4} = T_{abde}, T_{abdf} \] where
\[ T_{abde} = \{t_{19} \oplus \{e,o\}, t_{20} \oplus \{e,o\}\} = \{n\&mcof, k\&mcof\} = \{t_{29}, t_{30}\} \]
\[ T_{abdf} = \{t_{19} \oplus \{f,h\}, t_{20} \oplus \{f,h\}\} = \{n\&mceh, k\&mceh\} = \{t_{31}, t_{32}\} \]

To determine the trees differing from \( t_o \) by five edges we calculate the following sets
\[ S_f(t_{29}) \cap \xi(f) \oplus \{f\} = \{i,f,g,e\} \cap \{f,h\} \oplus \{f\} = \{\emptyset\} \]
\[ S_f(t_{30}) \cap \xi(f) \oplus \{f\} = \{i,f,g,e\} \cap \{f,h\} \oplus \{f\} = \{\emptyset\} \]
from which
\[ e_{i_1} e_{i_2} e_{i_3} e_{i_4} = \emptyset. \]

The set of all directed trees with base vertex 7 is
\[ T_7 = \{t_o, t_1, t_2, \ldots, t_{32}\}. \] This set of trees is shown in Figure 5.
Figure 5. Directed Trees of Ex. 2.
CHAPTER III

GENERATION OF DIRECTED SIMPLE PATHS WITHOUT DUPLICATION

3.1 Introduction

Methods of generating the set of directed paths, $P_{ij}'$, from vertex $i$ to vertex $j$ in a graph having $v$ vertices have been described in the literature \[4,5,10\]. These methods utilize the primitive connection matrix of the graph and obtain the paths by either evaluation of the permanent of the matrix, raising the matrix to the $v-1$ power or by applying a node pulling algorithm which transforms the matrix into a $2 \times 2$ matrix without altering the paths from $i$ to $j$. Each method generates the directed edge sequences from $i$ to $j$ from which the simple paths must be extracted without duplication. A further disadvantage when working with large graphs is that all paths are generated simultaneously making the quantity of data handled at each step very large. The method of raising the primitive connection matrix to the $v-1$ power generates all paths between every pair of vertices which provides more information than may be needed with a commensurate increase in effort.

The method to be described here generates the set of directed simple paths, $P_{ij}'$, of $G$ without duplication from a set of directed trees of the subgraph, $G'$. A minimal set of directed trees are generated and the paths are extracted from them.

It will first be shown that all paths in $P_{ij}$ are contained in the set of directed trees, $T_j$, of $G'$.
Theorem III: The set of directed simple paths, $P_{ij}$ of $G$, is equal to the set of directed simple paths of $G'$, where $G'$ is a sub graph of $G$ having vertices $\Omega(G') = \{w : w \in D^*(j)\}$ and edges $E(G') = \{e : e \in G$ and the initial and terminal vertices of $e$ are in $\Omega(G')\}$.

Proof: Assume \( \exists \) a path $p \subseteq G$ and $p \notin G'$. Then $p$ contains a vertex, $x$, $\exists x \in \Omega(G)$ and $x \notin \Omega(G')$. Hence $x \notin D^*(j)$, that is, $j$ is not accessible from $x$. Hence no path exists from $i$ to $j$ which contains $x$. Q.E.D.

It is evident that $G'$ is the maximum subgraph of $G$ which has a directed tree with base vertex $j$. Throughout the remainder of this chapter only the graph $G'$ will be considered so that the prime will be dropped.

Theorem IV: Every directed simple path in $P_{ij}$ of $G$ is contained in some tree, $t$, $\exists t \in T_j$, where $T_j$ is the set of directed trees of $G$ having $j$ as base vertex.

Proof: Given an arbitrary simple directed path, $p$, $\exists p \in P_{ij}$ we know that the set of vertices $\Omega(p)$ is a subset of $\Omega(G)$ $\exists j$ is accessible from $i$ in $p$. We use the notation $i \Rightarrow j$ to indicate that $j$ is accessible from $i$. A directed tree containing $p$ can be constructed by adding edges to $p$ until all vertices of $G$ are connected making sure that $\exists$ a path from each vertex to $j$.

Choose any arbitrary vertex, $y_1$, $\exists y_1 \in \Omega(G) - \Omega(p)$. By hypothesis $y_1 \Rightarrow j$ in $G$, hence $\exists$ a minimum set of edges, $M$, which completes a directed path from $y_1$ to a vertex in $\Omega(p)$. We form a
connected subgraph, \( H \), where \( E(H) = M \cup E(p) \). It is clear that \( \exists \) a directed path from every vertex of \( Q(H) \) to \( j \) and as \( M \) is a minimum set, no loops are contained in \( H \).

We now choose another arbitrary vertex, \( y_2 \), \( \exists y_2 \in \Omega(G) - \Omega(H) \). Again by hypothesis \( y_2 \rightarrow j \) so that \( \exists \) a minimum set of edges, \( M' \), which completes a directed path from \( y_2 \) to a vertex in \( \Omega(H) \). We form a connected subgraph, \( H' \), where \( H' \) contains the edges \( M' \cup M \cup E(p) \). \( H' \) includes more vertices of \( G \) than \( H \) and \( \exists \) a directed path from each vertex of \( H' \) to \( j \). No loops are present in \( H' \) as \( M' \) is a minimum set.

This procedure is continued until all vertices of \( G \) are connected. The resulting subgraph is a directed tree containing the path \( p \). We conclude that for any simple directed path, \( p \) in \( P_{ij} \) of \( G \), \( \exists \) a directed tree, \( t \) of \( T_j \) \( \exists t \supseteq p \). Q.E.D.

### 3.2 Obtaining the Path from a Tree

It follows from its definition that a directed tree, \( t \), of \( G \) having base vertex \( j \) contains exactly one directed path, \( p \), \( \exists p \in P_{ij} \).

Given a directed tree, \( t \), where \( t \in T_j \) of \( G \), the path \( p \) of \( P_{ij} \) can be obtained by elementary set operations by the procedure which follows.

Let \( I(i) = \{ e : e \in E(G) \text{ and the initial vertex of } e \text{ is } i \} \),

\[ \lambda(x) = \{ e : e \in E(G) \text{ and the initial vertex of } e \text{ is the terminal vertex of } x \text{ if the terminal vertex of } x \text{ is not } j \}, \]

\[ \lambda(x) = \{ \emptyset \text{ if the terminal vertex of } x \text{ is } j \} \],

\[ t = \{ e_1, e_2, \ldots, e_{v-1} \} \in T_j. \]
Consider a tree, \( t \), containing path, \( p \), in which \( j \) is the base vertex of \( t \) and the terminal vertex of \( p \). The path, \( p \), includes the edges 

\[
I(i) \cap t, \lambda[I(i) \cap t] \cap t, \lambda[\lambda[I(i) \cap t] \cap t] \cap t, \\
\lambda[\lambda[\lambda[I(i) \cap t] \cap t] \cap t] \cap t, \text{ etc. until } \lambda(x) = \emptyset \text{ which indicates that the path, } p, \text{ is complete.}
\]

\[
p = \prod_{j=1}^{k-1} (x_j)
\]

where \( x_j = \lambda(x_{j-1}) \cap t \) for \( 1 < j \leq k-1 \)

\[
= I(i) \cap t \text{ for } j = 1
\]

and \( k \) is the smallest integer \( \exists \lambda(x_k) = \emptyset \).

3.3 Generation of a Minimal Set of Trees Containing \( P_{ij} \)

The number of directed trees of a graph is often quite large. For example, the number of directed trees of the complete graph on \( v \) vertices, having the same base vertex is \([1]\)

\[
N_t = v^{v-2}.
\]

The number of directed simple paths in \( P_{ij} \) of \( G \) is at most equal to the number of directed trees in \( T_j \) and is usually much less; in general there are several directed trees containing the same directed path. For the case of the complete graph on \( v \) vertices the number of simple directed paths is

\[
N_p = \{\text{(number of paths of length one)} + \text{(number of paths of length two)} + \\
\ldots + \text{(number of paths of length } v-1)\}.
\]

\[
N_p = \{1 + (v-2) + (v-2)(v-3) + (v-2)(v-3)(v-4) + \ldots + (v-2)\!\}
\]

\[
N_p = (v-2)! \sum_{x=0}^{v-2} \frac{1}{x!}
\]

(3.3)
When it is desired to list the set of directed paths, \( P_{ij} \), in \( G \), it can be done by first generating the directed trees having \( j \) as base vertex and from these obtaining the distinct paths by the method described. One obvious disadvantage to this procedure is the vast number of directed trees which may result in some cases, while the number of directed simple paths is relatively small. For example, the number of directed trees of a complete graph of ten vertices is

\[
N_t = v^{v-2} = 10^8 = 100 \text{ million trees},
\]

while the number of directed simple paths in \( P_{ij} \) of the same graph is

\[
N_p = (v-2)! \sum_{x=0}^{v-2} \frac{1}{x!} = (10-2)! \sum_{x=0}^{8} \frac{1}{x!} = 109,601 \text{ paths}.
\]

In order to reduce the effort required to list all paths in \( P_{ij} \) of \( G \) it is desirable to generate a minimal set of directed trees each of which contains a distinct path, \( p, \exists p \in P_{ij} \), that is, we wish to generate one directed tree of \( G \) for each path in \( P_{ij} \) of \( G \). This procedure is now described.

**Theorem V:** A distinct directed tree of \( G \) containing a distinct directed simple path \( p \in P_{ij} \) of \( G \) can be formed from \( t_o \), a reference directed tree of \( G \), by replacing branch \( e_i \in t_o \) by an edge \( e \in G \) iff \( e \in S_{e_i}(t_o) \cap \xi(e_i) \oplus \{ e_i \} \), and \( e_i \in t_o \cap p_o \) where \( p_o \) is the directed simple path contained in \( t_o \).

**Proof:** By Theorem I we know that distinct trees are generated; we need only prove that each tree contains a distinct path.
Assume $e \in S(e_1 \cap \Sigma(e_1) \oplus \{e_1\})$ and $e_1 \in t_o \cap p_o$. Since $e_1 \in p_o$ the removal of $e_1$ from $t_o$ separates $t_o$ into two parts; vertex $i$ is in one part and vertex $j$ is in the other. The element, $e$, replacing $e_1$ forms a tree hence it completes a path from $i$ to $j$. Since $e$ is contained in the path in the new tree and each new tree has a distinct element, $e$, the paths are all distinct.

Assume $e_1 \notin t_o$ is replaced by $e \in G$ and a distinct tree, $t$, is formed containing a distinct path. Again it is known from Theorem I that $e \in S_e(t_o) \cap \Sigma(e) \oplus \{e\}$, hence we need only show that $e_1 \notin t_o \cap p_o$. If we assume that $e_1 \notin p_o$ then the new tree formed contains $p_o$. Since a directed tree contains exactly one path the new tree cannot contain a path distinct from $p_o$. This is a contradiction as, by hypothesis, $t$ contains a distinct path. We conclude that $e_1 \in p_o$. Q.E.D.

A set of distinct trees, $T_x e_1$, containing a set of distinct paths, $P x e_1$, is formed by repeating this replacement for each edge, $e \in S e_1(t_o) \cap \Sigma(e_1) \oplus \{e_1\}$. A class of these sets is formed by repeating the entire procedure for each $e_1 \in t_o \cap p_o$.

The number of trees in each set of $T x e_1$ will in general depend upon the choice of $t_o$. In order to generate the maximum number of distinct paths, it is necessary that $t_o$ be a directed tree such that the intersection, $S e_1(t_o) \cap \Sigma(e_1)$, is maximum for all $e_1 \in E(p)$. Any arbitrary $t_o$ can be modified to meet this requirement by a procedure described in section 3.4. If we let $t^* = \gamma(t_o)$ be a reference tree so
modified, the class of maximum sets can be expressed as

\[ T_{x}^{e_i} = \{ t : t = t_0^{*} \oplus \{ e_i, e \}, e_i \in t_0^{*} \cap p_0, p_0 \subseteq t_0^{*}, t_0^{*} = \gamma(t_0), \] \[ e \in S_{e_i}^{*}(t_0^{*}) \cap \xi(e_i) \oplus \{ e_i \} \}. \quad (3.4) \]

The corresponding set of paths, \( P_{x}^{e_i} \), contained in \( T_{x}^{e_i} \) is obtained in ordered form by the method described in section 3.2.

Additional sets of distinct trees of \( G \) containing distinct paths in \( P_{ij} \) can be formed from \( T_{x}^{e_i} \) by replacing edge \( e_i' \) of \( t \in T_{x}^{e_i} \) with edge \( e' \) where \( e_i' \) belongs to the path, \( p \), contained in \( t \) and \( e_i' \) is of higher order in \( p \) than \( e_i \) was in \( p_0 \). The edge \( e' \) must belong to the fundamental cut set, \( S_{e_i}(t) \), and issue from the same vertex as \( e_i' \).

This set will be maximized if the tree, \( t \), is replaced by \( t^{*} \), where \( t^{*} = \gamma(t) \). If we let \( x_1 \) be the order of \( e_i \) in \( p_0 \) and \( x_2 \) be the order of \( e_i' \) in \( p \) then for a class of maximum sets we can write

\[ T_{x_1 x_2}^{e_i e_i'} = \{ t' : t' = t^{*} \oplus \{ e_i', e_i \}, e_i' \in t^{*} \cap p, t^{*} = \gamma(t), \] \[ p \subseteq t^{*}, t \in T_{x}^{e_i}, e_i' \in S_{e_i}(t^{*}) \cap \xi(e_i') \oplus \{ e_i' \}, \] \quad (3.5) \]

where \( x_1 < x_2 < v-1 \).

The procedure can be continued to obtain a minimal set of distinct trees of \( G \) which contain the set of paths, \( P_{ij} \), of \( G \). The general expression for a class of distinct trees containing distinct paths is
\[ T_{x_1 x_2 \ldots x_k} = \{ t : t = t^* \bigoplus \{ e_i \}, e_i \in t^* \cap p, \]
\[ t^* = \gamma(t'), p \subseteq t^*, e \in S_{e_i} (t^*) \cap \xi(e_i) \bigoplus \{ e_i \}, \]
\[ t' \in T_{x_1 x_2 \ldots x_{k-1}} \]

where \( x_1 < x_2 < \ldots < x_k < v-1 \).

3.4 Modification of a Tree to Maximize \( S_{a_i} (t) \cap \xi(a_i) \) for a Given Path

The procedure given for the generation of all paths of \( G \) requires that at each step we take a tree, \( t \), containing a path, \( p \), where \( p = \{ a_1, a_2, \ldots, a_n \} \), and generate a new tree for each edge, \( e \), of \( \xi(a_i) \bigoplus \{ a_i \} \) if there is a path in \( G \) containing the first \( i-1 \) ordered edges of \( p \) and containing \( e \) as the \( i \)th edge. If \( \xi(a_i) \subseteq S_{a_i} (t) \) \( \forall a_i \in p \), we can form a new tree for each edge of \( \xi(a_i) \bigoplus \{ a_i \} \) by equation 3.6. This condition gives the maximum number of edges in the intersection, \( S_{a_i} (t) \cap \xi(a_i) \), and consequently gives the maximum number of trees which can be generated by equation 3.6. Maximization of the number of trees generated does not require that \( \xi(a_i) \subseteq S_{a_i} (t) \) as there may be edges in \( \xi(a_i) \) which do not appear as the \( i \)th edge in an ordered path in \( G \) containing the first \( i-1 \) ordered edges of \( p \); however, it is necessary that \( \xi(a_i) \cap S_{a_i} (t) \) contain all the edges of \( \xi(a_i) \) which do appear as the \( i \)th edge of a path in \( G \) which has as the first \( i-1 \) edges the corresponding edges of \( p \).
If $\xi(a_i) \not\subseteq S_{a_i}(t)$ we have the case where an edge $e \not\subseteq e \in \xi(a_i)$ terminates on a vertex from which the initial vertex of $a_i$ is accessible in $t$. If $\exists$ a distinct path, $p'$, in $G$ containing the first $i-1$ ordered edges of $p$ and containing $e$ as its $i^{th}$ edge then $\exists$ an edge, $c \in S_{a_i}(t) - \xi(a_i)$ which is also in $p'$. The edge, $c$, must have the same direction as $a_i$ in $S_{a_i}(t)$ hence it must be in the positive semi-cut, $S^+_{a_i}(t)$. A new tree, $t_1$, can be formed from $t$ by replacing the edge $x$ of $t$ with edge $c$ without altering the path, $p$, if $x = \xi(c) \cap t - p$.

$$t_1 = t \oplus \{c, x\}, \quad x = \xi(c) \cap t - p, \quad c \in S^+_{a_i}(t) - \xi(a_i)$$

(3.7)

$S^+_{a_i}(t)$ is the positive semi-cut of $G$ with respect to $t$ and contains $a_i$.

The edges of $E(G)$ in $S_{a_i}(t) \cap \xi(a_i)$ are also in $S_{a_i}(t_1) \cap \xi(a_i)$ as any vertex from which the initial vertex of $a_i$ is not accessible in $t$ is a vertex from which the initial vertex of $a_i$ is not accessible in $t_1$.

If the edge $e \not\subseteq S_{a_i}(t_1) \cap \xi(a_i) \oplus \{a_i\}$ then either $\exists$ $p'$ or $\exists$ an edge $c' \in S^+_{a_i}(t_1) - \xi(a_i)$ which $\not\subseteq p'$. If the latter is true we can form a tree, $t_2$, by replacing edge $x'$ of $t_1$ with $c'$ if $x' = \xi(c') \cap t_1 - p$.

$$t_2 = t_1 \oplus \{c', x'\}, \quad x' = \xi(c') \cap t_1 - p, \quad c' \in S^+_{a_i}(t_1) - \xi(a_i)$$

(3.8)

Continue this procedure until either $e \not\subseteq S_{a_i}(t_j) \cap \xi(a_i)$ or until no edge, $x$, exists. We are assured of reaching one or the other condition as each time the tree is modified the number of vertices from which the initial vertex of $a_i$ is accessible is reduced. The number of
vertices of $G$ is finite so that the procedure must terminate.

An edge $e \in E(G)$ which belongs to $\xi(a_i)$ also belongs to $S_{a_i}(t)$ if the initial vertex of $a_i$ is not accessible from the terminal vertex of $e$. When modifying a tree, $t$, with respect to $a_i$, according to equation 3.7, the set of vertices from which the terminal vertex of $a_i$ is not accessible in $t$ is contained in the set of vertices from which the terminal vertex of $a_i$ is not accessible in the modified tree, $t'$. Hence $\xi(a_i) \cap S_{a_i}(t)$ remains invariant under this operation and a tree may be modified so that the intersection $\xi(a_i) \cap S_{a_i}(t)$ is a maximum for each $a_i \in p$ if we start with the highest order edge in $p$ and proceed in reverse order to the lowest order edge. The resulting tree, $t^* = \gamma(t)$, is the tree required in equation 3.4.

The preceding can be summarized in the following theorem.

**Theorem VI:** Given an ordered simple directed path, $p = \{a_1, a_2, \ldots, a_n\}$, belonging to $P_{ij}$ of $G$ and a directed tree $t \supseteq p$, if $\exists$ an ordered simple directed path $p'$ in $G$ containing the first $i-1$ edges of $p$, then $e$, the $i^{th}$ element of $p'$, belongs to $\xi(a_i) \cap S_{a_i}(t^*)$ where $t^* = \gamma(t)$.

**Proof:** Assume $\exists$ a simple directed path $p'$. The $i^{th}$ edge of $p'$, $e$, must necessarily belong to $\xi(a_i)$. Let $t^* = \gamma(t)$ and assume $\exists x \in x = \xi(c) \cap t^* - p$ where $c \in S_{a_i}^+(t^*)$, then each edge of $S_{a_i}^+(t^*)$ issues from a vertex in $\Omega(p)$ from which the initial vertex of $a_i$ is accessible. If $e \notin S_{a_i}^+(t^*)$, it terminates on a vertex from which the initial vertex of $a_i$ is accessible in $t^*$ and $\exists$ an edge $c \in S_{a_i}^+(t^*) \ni c \in p'$. 
This means that a loop in $p'$ as the initial vertex of $e$ is accessible from the initial vertex of $c$ and vica versa. This is a contradiction as $p'$ is assumed to be a simple directed path. Hence $e \in \xi(a_i) \cap S_{a_i}(t^*)$. Q.E.D.

3.5 Verification that all Paths are Generated

**Theorem VII:** Every simple directed path in $P_{ij}$ of $G$ is contained in a directed tree generated by the procedure given in section 3.3.

**Proof:** Let $p$ be an arbitrary ordered simple directed path in $P_{ij}$ of $G$. Let $t_o$ be a reference directed tree of $G$ containing $p_o$ and let $a_i$ be the lowest ordered edge of $p_o$ which is not in $p$. From Theorem VI it is known that the $i^{th}$ edge of $p$ belongs to $\xi(a_i) \cap S_{a_i}(t^*)$ and by equation 3.6 a tree, $t_1$, can be generated from $t_o$ containing a path, $p_1$, which contains at least the first $i$ edges of $p$. Let $a'_k$ be the lowest order edge of $p_1$ which is not in $p$, $(i < k \leq n)$. Again from Theorem VI it is known that the $k^{th}$ edge of $p$ belongs to $\xi(a'_k) \cap S_{a'_k}(t^*_1)$ and by equation 3.6 a tree, $t_2$, can be formed containing a path, $p_2$, which contains at least the first $k$ edges of $p$. Continuing this reasoning we obtain the result that a tree is generated by the procedure given in section 3.3 which contains the path, $p$, and we conclude that every path in $P_{ij}$ of $G$ is generated. Q.E.D.
3.6 No Duplicate Paths are Generated

A path contained in the class of trees $T_{x_1x_2\ldots x_k}$ cannot be contained in the class $T_{y_1y_2\ldots y_k}$ unless

$$\{e_1,e_2,\ldots e_i\} = \{e_1,e_2,\ldots e_j\} \quad \text{and} \quad \{x_1,x_2,\ldots x_k = y_1,y_2,\ldots y_k\}. \quad (3.9)$$

This can be seen if you consider how a path in either class is formed. If in two given paths $e_1 \neq e_j$, then $x_1 \neq y_1$, and the number of leading ordered edges common to $p_o$ in the two paths is not the same. Hence, the two paths cannot be identical unless $e_1 = e_j$ and $x_1 = y_1$. If in two paths $e_1 = e_j$ but $e_i \neq e_j$ then if the $x_1$ th edge of each path is not the same the paths must be distinct, (the $x_1$ th edge is the edge replacing $e_1 = e_j$). If the $x_1$ th edge is the same then $e_i$ and $e_j$ belong to the same path $p_1$. In the two paths formed by replacing $e_i$ and $e_j$, one path will have the same leading ordered edges as $p_1$ up to the initial vertex of $e_i$ and the other will contain the same edges of $p_1$ up to the initial vertex of $e_j$. If $e_i \neq e_j$ the two paths must be distinct.

Hence $e_i = e_j$ if the two paths are identical and since $e_i$ and $e_j$ belong to the same path $p_1$, $x_2 = y_2$. Continuation of this reasoning leads to the conclusion that
if equations 3.9 and 3.10 are not satisfied.

Within a class of trees there are no duplicate paths because when replacing $e_i$ by edge $e$ in equation 3.6 a new path is generated for each distinct edge $e$. Each new path contains a different one of the distinct edges $e$, hence no duplicate paths are generated within the class of trees $T_{x_1x_2...x_k}$.

3.7 Example

The following example illustrates the procedure. Given the directed graph, $G$, of Figure 6, find all paths in $P^{15}$ of $G$. Let $t_o = \{agcd\}$ and $p_o = \{gc\}$. The reference tree is modified by computing the following sets of edges in $E(G)$.

$$\xi(c) - S_c(t_o) = \{b\} \quad S^+_c(t_o) - \xi(c) = \{n,e,h,f\}$$

![Figure 6. A Directed Graph](image-url)
\[ \xi(n) \cap t_0 - p_0 = \{n, h, a\} \cap \{a, g, c, d\} - \{g, c\} = \{a\} \]

\[ t_{o_1} = t_0 \oplus \{n, a\} = \{\text{ngcd}\} \]

\[ \xi(c) - S_c(t_{o_1}) = \{\emptyset\} \quad \xi(g) - S_g(t_{o_1}) = \{\emptyset\} \]

\[ t_{o_1} = \gamma(t_0) = t^*_o = \{\text{ngcd}\} \]

\[ T_{x_1} = \{T_1^g, T_2^c\} \]

\[ T_1^g = \{\&n c d, n f c d, e n c d\} = \{t_1, t_2, t_3\} \]

\[ T_2^c = \{g b n d, g k n d\} = \{t_4, t_5\} \]

We now modify the trees of \( T_{x_1} \) as follows.

\[ t_1 = \{\&n c d\} \text{ and } p_1 = \{\&n d\} \]

\[ \xi(d) - S_d(t_1) = \{m\} \quad S^+_d(t_1) - \xi(d) = \{g, f, a, h\} \]

\[ \xi(g) \cap t_1 - p_1 = \{e, f, g, \&\} \cap \{\&, n, d, c\} - \{\&, n, d\} = \{\emptyset\} \]

\[ \xi(f) \cap t_1 - p_1 = \text{same} \]

\[ \xi(a) \cap t_1 - p_1 = \{a, h, n\} \cap \{\&, n, d, c\} - \{\&, n, d\} = \{\emptyset\} \]

\[ \xi(h) \cap t_1 - p_1 = \text{same} \]

\[ \xi(n) - S_n(t_1) = \{\emptyset\} \quad \xi(\&) - S_\&(t_1) = \{\emptyset\} \]

\[ t_1 = t^*_1 = \gamma(t_1) = \{\&n d c\} \]

\[ t_2 = \{n f c d\} \text{ and } p_2 = \{f\} \]

\[ \xi(f) - S_f(t_2) = \{\emptyset\} \quad t_2 = t^*_2 = \{n f c d\} \]

\[ t_3 = \{e n c d\} \text{ and } p_3 = \{e d\} \]

\[ \xi(d) - S_d(t_3) = \{m\} \quad S^+_d(t_3) - \xi(d) = \{f, g, a, h\} \]
\(\xi(f) \cap t_3 - p_3 = \{e, f, g, \ell\} \cap \{e, n, c, d\} - \{e, d\} = \emptyset\)

\(\xi(g) \cap t_3 - p_3 = \text{same}\)

\(\xi(a) \cap t_3 - p_3 = \{a, h, n\} \cap \{e, n, c, d\} - \{e, d\} = \{n\}\)

\(t_{3,1} = t_3 \oplus \{a, n\} = \{eacd\}\)

\(\xi(d) - S_d(t_{3,1}) = \emptyset\)

\(t_{3,1} = t_3^* = \{eacd\}\)

\(t_4 = \{gbnd\} \text{ and } p_4 = \{gbnd\}\)

Since \(p_4 = t_4\), no modification is required and \(t_4 = t_4^* = \{gbnd\}\).

\(t_5 = \{gknd\} \text{ and } p_5 = \{gkd\}\)

\(\xi(d) - S_d(t_5) = \{j, m\}\)

\(s_d^+(t_5) - \xi(d) = \{f, h, c\}\)

\(\xi(f) \cap t_5 - p_5 = \{e, f, g, \ell\} \cap \{g, k, n, d\} - \{g, k, d\} = \emptyset\)

\(\xi(k) \cap t_5 - p_5 = \{h, n, a\} \cap \{g, k, n, d\} - \{g, k, d\} = \{n\}\)

\(t_{5,1} = t_5 \oplus \{h, n\} = \{gkhd\}\)

\(\xi(d) - S_d(t_{5,1}) = \{j\}\)

\(s_d^+(t_{5,1}) - \xi(d) = \{b, c, \ell, f\}\)

\(\xi(b) \cap t_{5,1} - p_5 = \{b, k, c\} \cap \{g, k, h, d\} - \{g, k, d\} = \emptyset\)

\(\xi(c) \cap t_{5,1} - p_5 = \text{same}\)

\(\xi(\ell) \cap t_{5,1} - p_5 = \{e, f, g, \ell\} \cap \{g, k, h, d\} - \{g, k, d\} = \emptyset\)

\(\xi(f) \cap t_{5,1} - p_5 = \text{same}\)

\(\xi(g) - S_g(t_{5,1}) = \emptyset\)

\(t_{5,1} = t_5^* = \{gkhd\}\)

\(e_{11} e_{12} = \{T_{12}^{gn}, T_{13}^{gd}, T_{12}^{gd}, T_{23}^{cn}, T_{23}^{cd}\}\)

\(T_{12}^{gn} = \{\elladc, \ellhdc\} = \{t_6, t_7\}\)

\(T_{13}^{gd} = \{\ellnjc\} = \{t_8\}\)
The trees of $T_{x_1 x_2}$ are now modified as follows.

$t_6 = [\lambda adc]$ and $p_6 = [\lambda ac]$

$\xi(c) - S_c(t_6) = \{b\}$

$S_c^+(t_6) - \xi(c) = \{e, f, h\}$

$\xi(e) \cap t_6 - p_6 = \{e, f, g, l\} \cap \{l, a, d, c\} - \{l, a, c\} = \{\emptyset\}$

$\xi(f) \cap t_6 - p_6 = \text{same}$

$\xi(h) \cap t_6 - p_6 = \{n, h, a\} \cap \{l, a, d, c\} - \{l, a, c\} = \{\emptyset\}$

$\xi(a) - S_a(t_6) = \{\emptyset\}$

$\xi(l) - S_l(t_6) = \{\emptyset\}$

$t_6 = t_6^* = [\lambda adc]$  

$t_7 = [\lambda hdc]$ and $p_7 = [\lambda h]$  

Since $t_7 \in T_{12}^{gn}$ and there are no edges of $p_7$ of order greater than two, $t_7$ cannot contribute any additional trees.

$t_8 = [\lambda njc]$ and $p_8 = [\lambda njc]$  

$t_8 = t_8^* = [\lambda njc]$  

$t_9 = [\lambda eac]$ and $p_9 = [\lambda emac]$  

$t_9 = t_9^* = [\lambda emac]$  

$t_{10} = [\lambda jac]$ and $p_{10} = [\lambda ejac]$  

$\xi(c) - S_c(t_{10}) = \{k, b\}$

$S_c^+(t_{10}) - \xi(c) = \{h, f, d\}$

$\xi(h) \cap t_{10} - p_{10} = \{h, a, n\} \cap \{e, j, a, c\} - \{e, j, c\} = \{a\}$

$t_{101} = t_{10} \oplus \{a, h\} = [\lambda hjc]$  

$\xi(c) - S_c(t_{101}) = \{k\}$

$S_c^+(t_{101}) - \xi(c) = \{l, f, m, d\}$

$\xi(l) \cap t_{101} - p_{10} = \{e, f, g, l\} \cap \{e, j, h, c\} - \{e, j, c\} = \{\emptyset\}$

$\xi(f) \cap t_{101} - p_{10} = \text{same}$
\( \xi(m) \cap t_{101} - p_{10} = \{m, d, j\} \cap \{e, j, h, c\} - \{e, j, c\} = \emptyset \)

\( \xi(d) \cap t_{101} - p_{10} = \text{same} \)

\( \xi(j) - S_j(t_{101}) = \emptyset \)

\( \xi(e) - S_e(t_{101}) = \emptyset \)

\( t_{101} = t^*_1 = \{e, j, c\} \)

\( t_{11} = \{gbd\} \) and \( p_{11} = \{gbh\} \)

\( t^*_{11} \) belongs to \( T_{23}^{cn} \) and as there are no edges of \( p_{11} \) of order greater than three, \( t^*_{11} \) cannot contribute any new trees.

\( t_{12} = \{gkmh\} \) and \( p_{12} = \{gkmh\} \)

\( t_{12} = t^*_{12} = \{gkmh\} \)

\( e^1_1 e^2_2 e^3_3 \)

\( T_{x_1 x_2 x_3} = \{T_{123}^{gnc}, T_{123}^{gda}, T_{123}^{gdc}\} \)

\( T_{123}^{gnc} = \{ladk\} = \{t_{13}\} \)

\( p_{13} = \{ladk\} \)

\( T_{123}^{gda} = \{emhc\} = \{t_{14}\} \)

\( p_{14} = \{emh\} \)

\( T_{123}^{gdc} = \{ejbh\} = \{t_{15}\} \)

\( p_{15} = \{ejbh\} \)

\( P_{ij} = \{p_0, p_1, p_2, \ldots, p_{15}\} \)

\( P_{ij} = \{gc, \&nd, f, ed, gbnd, gkd, \&ac, \&h, \&njc, emac, ejc, gbh, gkmh, \&akd, \)

\( \text{emh, ejbh}\} \)

A minimal set of trees containing all paths in \( P_{ij} \) is the set \( \{t_0, t_1, t_2, \ldots, t_{15}\} \).
CHAPTER IV
GENERATION OF DIRECTED 2-TREES WITHOUT DUPLICATION

4.1 Introduction

A 2-tree of a graph, G, is a pair of unconnected, circuitless subgraphs, each subgraph being connected, which together include all the vertices of \( \Omega(G) \) [11]. A directed 2-tree of G is a 2-tree in which each of the two subgraphs is a tree directed to a base vertex of that subgraph [3]. The symbol, \( T_{2d}^{k:j} \), is used to designate the set of 2-trees of G which have the property that vertex k is the base vertex for one part and vertex j is the base vertex for the other part.

It is often desirable to specify in which part of a directed 2-tree certain vertices, other than the base vertices, are located. The set of 2-trees having base vertices k and j with the additional requirement that vertex i be contained in the same subgraph as k and vertices q and r be contained in the same subgraph as vertex j is designated by the symbol \( T_{2d}^{i,k:q,r,j} \).

The application of directed 2-trees to the solution of network problems has been considered in the literature [1,3]. The following describes a method of generating two types of directed 2-trees without duplication. This method can be extended to other types of 2-trees.

4.2 Generation of the Set of 2-trees, \( T_{2d}^{k:j} \)

The directed 2-trees of G having base vertices k and j can be generated without duplication by the following procedure. First remove
the edges of $E(G)$ which issue from vertex $k$ and add an edge, $e$, which issues from $k$ and terminates on $j$. Next choose a reference directed tree, $\tau_0$, of the resulting graph, $G_1$, and generate the set of all directed trees, $T_{j1}$, of $G_1$, having base vertex $j$, by the procedure given in section 2.2. Finally, form a 2-tree from each tree of $T_{j1}$ by removing edge $e$, thus obtaining the set of all distinct 2-trees, $T_{2d_{k:j}}$.

$$T_{2d_{k:j}} = \{t_{2d_{k:j}} : t_{2d_{k:j}} = \tau \oplus \{e\}, \tau \in T_{j1}\}$$

(4.1)

**Theorem VIII**: The set of all directed 2-trees, $T_{2d_{k:j}}$, is generated by the procedure.

**Proof**: If $t_{2d_{k:j}}$ is an arbitrary 2-tree of the set $T_{2d_{k:j}}$, then there are two disjoint subsets of $\Omega(G)$, $A$ and $B \ni$ vertex $k$ is accessible in $t_{2d_{k:j}}$ from every vertex of $A$ and vertex $j$ is accessible in $t_{2d_{k:j}}$ from every vertex of $B$. It is also true that in this arbitrary 2-tree there is an edge issuing from vertex $k$ because $A$ and $B$ are disjoint and there are no loops in a 2-tree. Hence, only vertices and edges $\in G_1$ can appear in $t_{2d_{k:j}}$. The addition of edge, $(k,j)$, forms a directed tree, $\tau$, of $T_{j1}$. According to Theorem II, the procedure given in section 2.2 generates all directed trees of $G_1$, hence, it generates $\tau$, the tree from which $t_{2d_{k:j}}$ would be obtained by the procedure of this section. Therefore, it is concluded that all directed 2-trees are generated.

Q.E.D.
Theorem IX: No duplicate directed 2-trees are generated by the procedure given.

Proof: From section 2.6 it is known that the set of trees $T_{j1}$ are all distinct. Since each tree in the set contains edge, $(k,j)$, the removal of $(k,j)$ from each tree in $T_{j1}$ leaves only distinct directed 2-trees. Q.E.D.

4.3 Generation of the Set of 2-trees, $T_{2d_i,k:j}$

The directed 2-trees of $G$ having base vertices $k$ and $j$ with vertex $i$ in the same part as vertex $k$ can be generated without duplication as follows. Remove the edges of $E(G)$ which issue from vertex $k$ and add edge $e \ni e = (k,j)$, as in section 4.2, to form graph $G_1$. Next remove the edges of $E(G_1)$ which issue from vertex $i$ and add edge $e' \ni e' = (i,k)$ to form graph $G_2$. Choose a reference directed tree, $\tau'$, of $G_2$ and generate the set of all directed trees, $T_{j2}'$, of $G_2$ having $j$ as base vertex. Add edge $e'$ to graph $G_1$ to form $G_3$. Each directed tree, $\tau'$, of $T_{j2}'$ is also a directed tree of $G_3$ and a particular subset of directed trees of $G_3$ can be formed by replacing $e'$ in each tree of $T_{j2}'$ with an edge $\alpha$, where $\alpha \in \mathcal{E}'(e') \cap \mathcal{S}'_e(\tau') - \{e'\} \cup \mathcal{S}'_e(\tau')$. If no such $\alpha$ exists, $\tau'$ is discarded. The primed functions indicate that the operation is carried out on the edges of $G_3$. The particular subset obtained is a subset of directed trees, $W_{j1}$, of $G_1$ each containing the edge, $e$.

\[
W_{j1} = \{\tau : \tau = \tau' \oplus \{e',\alpha\}, \tau' \in T_{j2}', \\
\alpha \in \mathcal{E}'(e') \cap \mathcal{S}'_e(\tau') - \{e'\} \cup \mathcal{S}'_e(\tau'), \alpha exists; \\
\tau = \emptyset if \exists \alpha\} \tag{4.2}
\]
The desired set of 2-trees of $G$ is obtained by deleting edge, $e$, from each tree of $W_{ij}$.

$$T_{2d_{i,k,j}} = \{ t_{2d_{i,k,j}} : t_{2d_{i,k,j}} \oplus \{ e \}, \tau \in W_{ij} \} \quad (4.3)$$

**Theorem X:** All directed 2-trees of the set $T_{2d_{i,k,j}}$ are generated by the procedure given.

**Proof:** If $t_{2d_{i,k,j}}$ is an arbitrary directed 2-tree of $T_{2d_{i,k,j}}$, then find an edge in $t_{2d_{i,k,j}}$ issuing from vertex $k$ because $k$ is the base vertex for one part of $t_{2d_{i,k,j}}$ and there are no loops in a 2-tree. Addition of edge $e \ni e = (k,j)$ forms a directed tree, $\tau$, of $G_j$. From the definition of $t_{2d_{i,k,j}}$, it is known that $\exists$ one edge, $\beta$, issuing from vertex $i$ which terminates on a vertex from which $k$ is accessible in the 2-tree. By replacing $\beta$ with edge, $e'$, a tree, $\tau'$, of $G_j$ is obtained and the edge, $\beta$, does not belong to the fundamental cut-set, $S_e(\tau')$.

By Theorem II it is known that the given procedure generates the tree, $\tau'$, from which $t_{2d_{i,k,j}}$ is obtained by equations 4.2 and 4.3. Hence, we conclude that all directed 2-trees of $T_{2d_{i,k,j}}$ are generated. Q.E.D.

**Theorem XI:** No duplicate directed 2-trees of $T_{2d_{i,k,j}}$ are generated by the procedure.

**Proof:** Duplicate directed 2-trees could not be formed from a single tree, $\tau'$, of $G_j$ by replacing $e'$ with the edge $\alpha$, according to equations 4.2 and 4.3, because only distinct edges are used to replace $e'$. 
Hence, if two identical directed 2-trees are generated, they must result from two identical trees in $T_{j2}$, the set of directed trees of $G_2$. The procedure used to generate $T_{j2}$ is that of section 2.2 which does not produce duplicate directed trees. Therefore, no duplicate directed 2-trees can be generated by the procedure. Q.E.D.

4.4 Example

Given the directed graph of Figure 7, find all 2-trees of the set $T_{2d_{3,4;2}}$.

The graph $G_2$ is formed by deleting edges $\{q, b, p, c, m, n\}$ and adding edges $\{e, e'\}$, where $e = (4, 2)$ and $e' = (3, 4)$. The set of all directed trees of $G_2$ having vertex 2 as base vertex is generated by the procedure of section 2.2 and is the set $T_{22} = \{aehe', fehe', aege', aede', fege', fede'\} = \{\tau_1', \tau_2', \tau_3', \tau_4', \tau_5', \tau_6'\}$.

The graph, $G_3$, is formed from $G$ by deleting edges $\{c, m, n\}$ and adding edges $\{e, e'\}$. The following sets are now computed.
\[ S'(e') \cap S'_e(\tau_1^l) - S'_e(\tau_1^l) \cup \{e'\} = \{b, q, p, e'\} \cap \{q, g, f, b, e'\} \]
- \{c, d, e\} \cup \{e'\} = \{b, q\}

\[ S'(e') \cap S'_e(\tau_2^l) - S'_e(\tau_2^l) \cup \{e'\} = \{q, b, p, e'\} \cap \{a, b, g, q, p, e'\} \]
- \{c, d, e\} \cup \{e'\} = \{b, p, q\}

\[ S'(e') \cap S'_e(\tau_3^l) - S'_e(\tau_3^l) \cup \{e'\} = \{q, b, p, e'\} \cap \{b, h, d, e'\} \]
- \{c, d, e\} \cup \{e'\} = \{b\}

\[ S'(e') \cap S'_e(\tau_4^l) - S'_e(\tau_4^l) \cup \{e'\} = \{q, b, p, e'\} \cap \{b, g, q, f, e'\} \]
- \{c, h, g, q, f, e\} \cup \{e'\} = \{b\}

\[ S'(e') \cap S'_e(\tau_5^l) - S'_e(\tau_5^l) \cup \{e'\} = \{q, b, p, e'\} \cap \{b, h, d, e'\} \]
- \{c, d, e\} \cup \{e'\} = \{b\}

\[ S'(e') \cap S'_e(\tau_6^l) - S'_e(\tau_6^l) \cup \{e'\} = \{q, b, p, e'\} \cap \{b, g, q, a, p, e'\} \]
- \{c, h, g, a, p, q, e\} \cup \{e'\} = \{b\}

The required subset, \( W_{21} \), of directed trees of \( G_1 \) is obtained by equation 4.2.

\[ W_{21} = \{abhe, aqhe, fbhe, fphe, fqhe, abge, abde, fbge, fbde\} \]

The set of directed 2-trees of \( G \) is obtained by equation 4.3.

\[ T_{2d_{3,4:2}} = \{abh, aqf, fbh, fph, fqh, abg, abd, fbg, fbd\} \]
CHAPTER V

CONCLUDING REMARKS

The application of graph theory to the solution of a vast number of problems has been demonstrated by many authors. Directed graphs have been employed in the analysis of electrical networks [11], communication networks [7], switching circuits [5], and even the simultaneous solution of linear equations [2]. Evaluation of network determinants and cofactors by directed trees and 2-trees has been demonstrated by Chen [1] and Dodd [3]. Signal flow graphs and flow graphs, which were first introduced by Mason [6] and Coates [2], are directed graphs which have been applied to the analysis of problems in feedback theory, network stability, matrix inversion and others [1]. Because the solutions frequently require the set of directed trees, 2-trees or paths of an associated directed graph, it is desirable and often necessary that an economical method of generating these be available.

Procedures for generating the directed trees, 2-trees and paths of a directed graph have been presented in this paper. These procedures generate no duplicates thus avoiding the necessity of repeated search to eliminate the duplicate elements. Only elementary set operations are involved and the procedures are readily implemented by a digital computer.
BIBLIOGRAPHY


VITA

Archie Joseph Paul, Jr. was born on August 10, 1927 in Crivitz, Wisconsin. He received the degree of B.S. (E.E.) from the University of Wisconsin in 1950 and the degree of M.E.E. from Syracuse University in 1961.

He was employed by the International Business Machines Corporation from 1950 through 1961. He held an IBM Resident Study Fellowship from 1962 to 1965. Since September, 1963 he has been a Research Fellow of the Coordinated Science Laboratory at the University of Illinois.

Mr. Paul is a member of Tau Beta Pi, Eta Kappa Nu and Phi Eta Sigma.
<table>
<thead>
<tr>
<th>Distribution list as of September 1, 1964</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Distribution Air Force Technical Library Maxwell Air Force Base, Alabama 36112-5000</td>
</tr>
</tbody>
</table>
| **1.** Materials Research Laboratory 
US Army Research and Development 
Washington, D.C. 20302 |
| **1.** Distribution Material Research Laboratory 
US Army Research and Development 
Washington, D.C. 20302 |
| **2.** National Aeronautics and Space Administration 
Washington, D.C. 20546 |
| **3.** Defense Intelligence Agency 
Washington, D.C. 20511 |
| **4.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **5.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **6.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **7.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **8.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **9.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **10.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **11.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **12.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **13.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **14.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **15.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **16.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **17.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **18.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **19.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **20.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **21.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **22.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **23.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **24.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **25.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **26.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **27.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **28.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **29.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **30.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **31.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **32.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **33.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **34.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **35.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **36.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **37.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **38.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **39.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **40.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **41.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **42.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **43.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **44.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **45.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **46.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **47.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **48.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **49.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |
| **50.** Air Force Office of Scientific Research 
Office of the Secretary of Defense 
Washington, D.C. 20301 |