### STRUCTURAL SYNTHESIS AND ANALYSIS

# OF KINEMATIC CHAINS USING

### PATH MATRICES

By

FREDERICK FENN QUIST, JR.

United States Military Academy

West Point, New York

1964

Submitted to the Faculty of the Graduate College of the Oklahoma State University in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE May, 1971 이 너희 나에서 제외에 지갑한 사람이 나지? 가지 않네.

example in the second statistical second

2003/2022 (C20)

Thesis 1971 Q85 Cop. 2

sound for the second second

Bartest drawn (Cillean) Arnian

184.

Substant in lts Xutuity of the destains, Subserve of real Definitions in [Linears) net to Definitions of the line choice subserve MARTER of the second line choice



# Thesis Approved:

<u>Manrom</u> <u>M</u> Thesis Advi

Dean of the Graduate College



### PREFACE

This thesis presents a method of describing kinematic chains and mechanisms using matrices which represent paths or circuits in a chain. The path matrices developed permit the use of the digital computer in developing mechanisms from kinematic chains, synthesizing modified chains based on basic kinematic chains, and comparing kinematic chains. Computer programs using the path matrices were applied to the 230 tenlink chains and resulted in tremendous amounts of data which are summarized in the body of the thesis.

I would like to acknowledge my indebtedness to my adviser, Dr. A. H. Soni, whose inspiration and support were instrumental in the development of this thesis.

Indebtedness is also acknowledged to the National Science Foundation which furnished part of the finances for this research under NSF Grant GU-3160.

Finally, I wish to thank my wife and daughter for their patience, understanding, and support.

iii

# TABLE OF CONTENTS

Chapte	r	Page
I.	INTRODUCTION	1
11.	DEFINITIONS OF NOTATION AND TERMS	4
111.	DEVELOPMENT OF THE CHAIN PATH MATRIX	9
IV.	DEVELOPMENT AND USE OF THE MECHANISM PATH MATRIX	18
	Theory	18 21 23
	Mechanisms	23
<b>v</b> .	CHAINS WITH SPRINGS	26
	Theory	26
	With Springs	30
VI.	CHAINS WITH PULLEYS AND BELTS	34
	Theory	34
	Ainematic Chains to Chains with Pulleys and Belts	37
VII.	CHAINS WITH CAM PAIRS	41
VIII.	CHAINS WITH PRISM PAIRS	46
IX.	CHAINS WITH DOUBLE JOINTS	52
	Theory.	52
	With Double Joints	56
х.	COMPARISON OF MECHANISMS DERIVED FROM CHAINS WITH SPECIAL ELEMENTS	60
	Example 1	60 65 69

۰.

Chapter		Page
XI. STRUCTU	RAL SYNTHESIS OF GEAR TRAINS	73
XII. SUMMARY	AND CONCLUSIONS	79
A SELECTED BIB	LIOGRAPHY	82
APPENDICES	• • • • • • • • • • • • • • • • • • • •	83
APPENDIX APPENDIX	A - LINE DRAWINGS OF THE 230 TEN-LINK CHAINS B - LOOP MATRICES FOR TEN-LINK CHAINS WITH	84
	CROSSED LINKS	90
APPENDIX	C - SUBROUTINE LOOP	93
APPENDIX	D - SUBROUTINE MECH	98
APPENDIX APPENDIX	E - SUBROUTINE MECCOM	101
<b>A DDRNINT Y</b>	CHAINS.	106
ADDENIDTY	CHAINS WITH SPRINGS	111
AFTENDIA	TEN_LINK CHAINS WITH PULLEYS AND	
ADDENDTY	BELTS	121
APPENDIX	TEN-LINK CHAINS WITH CAM PAIRS	128
APPENDIX	J - SUBROUTINES USED IN DETERMINING THE TEN-LINK CHAINS WITH PRISM PAIRS	133
APPENDIX	K - SUBROUTINES USED IN DETERMINING THE	
	TEN-LINK CHAINS WITH DOUBLE JOINTS	138

v

### LIST OF TABLES

Table		Page
I.	Loop Matrix for Chain 189	12
II.	Combinations of Loops Used in Forming the CPM	12
111.	<b>CPM</b> for Chain 189	13
IV.	Computer Loop Matrix for Chain 189	13
V.	Row Combinations Used in Forming the CPM for Ten-Link Chains	17
VI.	Computer CPM for Chain 189	20
VII.	MPM for Mechanism 1 of Chain 189	20
VIII.	MPM of Table VII Reduced to Matrix of Link Types	20
IX.	MPM for Mechanism 5 of Chain 189	22
Х.	Sample Output for Ten-Link Mechanisms	24
XI.	MPM for Chain 189 with Links 3 and 4 Replaced by a Spring	29
XII.	MPM for Chain 189 with Links 7 and 8 Replaced by a Spring	29
XIII.	Modified MPM for Chain 189 with Two Springs	29
XIV.	Equivalence of Rows with Two Springs	29
XV.	Sample Output for Chain 189 with Springs	32
XVI.	Ten-Link Chains with Springs	33
XVII.	MPM for Chain 189 with Link 1 Converted to a Pulley and Links 8 and 9 Converted to a Belt	36
XVIII.	Sample Output for Chain 189 with Pulleys	39
XIX.	Ten-Link Chains with Pulleys	40
XX.	Sample Output for Chain 189 with Cam Pairs	43

Table		Page
XXI.	Ten-Link Chains with Cam Pairs	45
XXII.	Sample Output for Chain 189 with Prism Pairs,	48
XXIII.	Ten-Link Chains with Prism Pairs	51
XXIV.	MPM for Chain 189 with Double Joint at Position 1(8,9)	55
XXV.	Sample Output for Chain 189 with Double Joints	57
XXVI.	Ten-Link Chains with Double Joints	59
XXVII.	LOOP Matrix for Chain 4	62
XXVIII.	LOOP Matric for Chain 4 with Special Elements	62
XXIX.	CPM for Chain 4 with Special Elements	62
xxx.	MPM's for Chain 4 with Special Elements	64
XXXI.	LOOP Matrix for Chain 48 with Special Elements	67
XXXII.	CPM for Chain 48 with Special Elements	67
XXXIII.	MPM's for Chain 48 with Special Elements	68
XXXIV.	LOOP Matrix for Chain 206	71
xxxv.	LOOP Matrix for Chain 206 with Special Elements	71
XXXVI.	CPM for Chain 206 with Special Elements	71
XXXVII.	MPM's for Chain 206 with Special Elements	72
XXXVIII.	Mandatory Base Points and Rotation Points • • • • • • • •	74
XXXIX.	Rules Applying to Mandatory Points in Figure 18	74
XL.	LOOP Matrices for Group 64D of Appendix A	91
XLI.	Identification of Variables Used in Subroutine LOOP	94
XLII.	Subroutine LOOP	95
XLIII.	Subroutine MECH	99
XLIV.	Variables Used in Subroutine MECCOM	102
XLV.	Subroutine MECCOM	103

· · ...

XLVI. Program for Determining Ten-Link Mechanisms . . . . . . 107

Table		Page
XLVII.	Input Data for Computer Programs	109
XLVIII.	Program for Determining Ten-Link Chains with Springs	112
XLIX.	Subroutine PULLOC	122
L.	Subroutine PUL3	125
LI.	Subroutine CAMLOC	129
LII	Subroutine CAM3	131
LIII.	Subroutine JOINT	134
LIV.	Subroutine PRISM2	136
LV.	Subroutine DBLOC	139
LVI.	Subroutine DBL2	141
LVII.	Subroutines LNKCNT and MOBCK	144

.

# LIST OF FIGURES

Figu	ire	Page
1.	Representation of Kinematic Links	5
2.	Representation of Kinematic Joints	5
3.	Six-Link Basic Kinematic Chain	5
4.	Ten-Link Mechanism	5
5.	Four Loop Kinematic Chain	7
6.	Ten-Link Chain Number 189	10
7.	Two Binary Links Replaced by a Spring,	27
8.	Ternary Link and Two Binary Links Converted to a Pulley and Belt	35
9.	Ternary Link with 3 Binary Links Converted to Pulleys	38
10.	Ternary Links Sharing a Binary Link Converted to Pulleys and Belt	38
11.	Binary Link Converted to a Cam Pair	42
12.	Addition of Double Joint to Basic Kinematic Chain	53
13.	Demonstration of Rule 1 for Double Joints	53
14.	Demonstration of Rule 2 for Double Joints	53
15.	Chain 4 with Special Elements Added	61
16.	Chain 48 with Special Elements Added	66
17.	Chain 206 with Special Elements Added	70
18.	Chain 223 with Double Joint and Base and Rotation Points .	76
19.	Equivalent Gear Linkage for Figure 18	76
20.	Line Drawings of Ten-Link Chains	85
21.	Flow Chart for Subroutine LOOP	97

Figu	re															Page
22.	Flow	Chart	for	Subroutine	MECH	•	•••	•	•	• •	•	•	• •	•	• •	100
23.	Flow	Chart	for	Subroutine	MECCOM.	•	••	•	•	• •	•	•	• •	•	•	105
24.	Flow	Chart	for	Subroutine	SPRLOC.	•	••	•	•	••	•	•	•	• •	•	119
25,	Flow	Chart	for	Subroutine	SPRNG3.	•	••	•	•	••	•	•	•	•	•	120
26.	Flow	Chart	for	Subroutine	PULLOC.	•		•	• .	••	. •	•	• •	•	• s	124
27,	Flow	Chart	for	Subroutine	PUL3	•	• •	•	•	• .•	•	•	• •	•	•	127
28.	Flow	Chart	for	Subroutine	CAMLOC.	•	••	•	. •	••	•	•	• •	•	•	130
29.	Flow	Chart	for	Subroutine	CAM3	• .	• •	•	•	••	•	• .	• •	•	• .	132
30.	Flow	Chart	for	Subroutine	JOINT .	•	••	• •	•	••	•	•	•	• . •	•	135
31.	Flow	Chart	for	Subroutine	PRISM2.	•	••	•	•	• •	•	•	• •	•	•	137
32.	Flow	Chart	for	Subroutine	DBLOC .	•	• .•	•	. •	••	•	•	• •	•	•	140
33.	Flow	Chart	for	Subroutine	DBL2	•	•••	•	•	••	. •	•	•	• •	•	143
34,	Flow	Charts	foi	Subroutine	es LNKCNT	C a	nd	MOI	зск	••	•	• .	•		•	145

Constant and the second

#### CHAPTER I

### INTRODUCTION

During the past several years, the problems of structural synthesis and analysis of kinematic chains have been approached in several ways. Davies and Crossley (1)\* utilized Franke's condensed notation, a method of representing kinematic chains as molecules, to obtain a census of seven, nine, ten, and eleven-bar kinematic chains. Hain (2,3,4,5) has done a great deal of work in this area of kinematics, working mainly with the six and eight-link chains. He has obtained censuses not only for the basic chains, but also for these chains with modifications such as the addition of cam pairs, prism pairs, and multiple joints. In addition, he has obtained the mechanisms derived from these chains and presented them schematically thus furnishing the designer with a large choice of solutions to a given problem. This work has been done mainly by visual inspection of the structural drawing of each chain.

The results obtained by the above individuals have relied heavily on the researcher's ability to visualize the various configurations a chain may assume and to recognize isomorphisms which may occur. As more complex chains are examined, this ability is heavily taxed and the tremendous number of chains and mechanisms makes a manual approach

\*Numbers in parentheses refer to numbered references in the Bibliography.

impractical. Therefore, the need arises for an automated method of performing the structural analysis of kinematic chains.

Woo (6) has used graph theory in enumerating the plane kinematic chains with 13 revolute pairs and 10 links, thus verifying some of the results obtained by Davies and Crossley above. An incidence matrix was used to develop the chains and also served as the structural descriminant to detect isomorphisms. Similarly, Dobrjanskyj and Freudenstein (7) have developed an algorithm which applies graph theory to the comparison of kinematic chains and mechanisms. Buchsbaum (8) has further developed the use of graph theory for the structural classification and type synthesis of chains with multiple elements.

All of the preceding approaches which utilize graph theory have the advantage of being computerized and have contributed significantly to the analysis of kinematic chains.

The purpose of this thesis is to present an alternate approach to the problem of type synthesis and analysis of kinematic chains which does not rely on graph theory but utilizes a method of comparing paths or circuits in a chain. A Chain Path Matrix (CPM) is developed and from it, the Mechanism Path Matrix (MPM) is formed. The MPM serves as the structural discriminant in comparing mechanisms and chains. The CPM is useful in converting basic kinematic chains into chains with special elements such as pulleys and belts, cams, double joints, etc. The method is easily computerized and, consequently, fills the need for automated methods of processing chains with more than eight links. Furthermore, it has the advantage of requiring no mathematics in its derivation or application.

The path matrix approach has been applied to the analysis of the

230 ten-link chains with one degree of freedom (Appendix A). The amount of data generated by this anaylsis is enormous, and, consequently, has not been included in this thesis. The data has, however, been included in reference (9), a copy of which is available in the Oklahoma State University Library and summaries of this data are included at the end of each appropriate chapter herein.

3

In the chapters which follow, the development of the CPM and MPM will be described. The application of these matrices in synthesizing mechanisms, developing chains with special elements (cams, springs, etc.) and converting kinematic chains to gear trains will then be discussed.

### CHAPTER II

### DEFINITIONS OF NOTATION AND TERMS

In order to develop a basis for the following chapters, a brief discussion of notation and terms will be presented.

A kinematic link is a rigid, massless member and is classified according to the number of joints which it can accept. A binary link is a link which can accept two joints; a ternary link, three joints; and a k-nary link, k joints. The standard kinematic notation used to represent various links is illustrated in Figure 1.

A kinematic pair consists of two contacting elements and is used to connect two kinematic links. Three types of kinematic pairs will be considered here:

1.	Revolute Pair	<ul> <li>Permits only relative rotation between two connected links. Also called a tu ing pair or pin joint.</li> </ul>	irn-
2.	Prism Pair	<ul> <li>Permits only relative translation betw two connected links. Also called a sl ing pair.</li> </ul>	/een .id-
3.	Cam Pair	- Permits both rotary and translatory mo tion between two connected links.	) —

In addition to the three pairs mentioned above, the double joint will also be used as a means of connecting kinematic links. The double joint is simply the merging of two revolute pairs to form a joint which permits rotational motion among three links.

The kinematic notation used to represent the various joints described above is illustrated in Figure 2.



5

(d) Double Joint





Figure 3. Six-Link Basic Kinematic Chain



Figure 4. Ten-Link Mechanism

A kinematic chain is constructed using kinematic links and pairs to form one or more closed loops. Each link must be free to move relative to the links connected to it (e.g., a structure such as three binary links joined by three revolute pairs to form a triangle is not a kinematic chain). If a kinematic chain is constructed so that the motion of all links takes place in one plane or in parallel planes, the chain is called a planar kinematic chain. A basic kinematic chain is a planar kinematic chain with all links connected by revolute pairs (Figure 3).

A kinematic mechanism is defined as a kinematic chain which has one link fixed with respect to ground and which has a single degree of freedom. The degrees of freedom of a basic kinematic chain (and the mechanisms derived from it) can be determined from the formula  $F = 6(N-1)-5P_1$  where F is the degrees of freedom, N is the number of links, and  $P_1$  is the number of revolute pairs. Basically, a mechanism (or chain) with one degree of freedom is one in which the motion of all links is constrained, that is, no link is free to move independently of the other links. Figure 4 illustrates a ten-link mechanism.

A path, as used here, will be defined as a sequence of connected elements encountered in traversing a chain (or mechanism) beginning at a given element and returning to that element without repeating any element. For example, Figure 5 illustrates a kinematic chain with four loops. The links have been numbered as a means of identification. One path beginning at link 1 could be described by the sequence of link numbers 1,2,10,9. This path represents the links encountered in traversing loop 1 in a clockwise direction. A second path can be represented by the sequence 1,2,10,5,6,7,8 and can be thought of as the path





generated by combining loops 1, 3, and 4. Several other paths may also be found which originate at link 1. Each one of these paths can be thought of as a combination of either loop 1 or loop 4 (loops which contain link 1) and one or more of the other loops of the chain. This point will be further illustrated in the next chapter.

The foregoing provides sufficient background for an understanding of the path comparison method. The following chapters will describe the development of the path matrices and their applications.

#### CHAPTER III

### DEVELOPMENT OF THE CHAIN PATH MATRIX

In the previous chapter, a path was defined as the sequence of elements encountered in traversing a loop or combination of loops in a kinematic chain. In this chapter, the development of the Chain Path Matrix (CPM), which represents all possible paths in a chain, will be discussed and a computer program for generating the CPM will be described.

In order to derive any meaning from the numbers representing the links in a chain, it is desirable to be able to designate both the link type (binary, ternary, quaternary, etc.) and the link number. To do this, a three or four digit integer will be used in which the ones and tens places represent the link number and the hundreds and thousands places represent the link type. For example, link number 1 of chain 189 (Figure 6) is a ternary link and this fact will be reflected by placing a three in the hundreds place of its identifying number. Since it is also link number 1, the digits 01 will be placed in the tens and ones positions. The identifying number for link 1 will then be 301. Similarly, the identifying number for link 2 is 402 and for link 3 is 203. The thousands place of the identifying number will be used in later chapters to designate special elements inserted in the basic kinematic chain.

The development of the CPM is based on a matrix called the loop

g



Figure 6. Ten-Link Chain Number 189

matrix which represents the path around each loop of the chain. The path is developed in a clockwise direction as used here although the direction is immaterial as long as all paths are taken in the same direction. Using the identifying numbers described above, the loop matrix for chain 189 would appear as shown in Table I. The starting element for each row may be chosen at random since the beginning and end elements of each row are considered to be adjacent.

The CPM will be composed of all possible combinations of the basic loops described by the loop matrix. The CPM will then be formed as shown in Table II and will identify every path or circuit which can be formed in the kinematic chain. In order for two loops to combine, they must have at least two elements in common, otherwise, an element in the combined path would be repeated. Because of this restriction, some of the combinations shown in Table II cannot be realized. When this case occurs, the row of the CPM representing that combination will be filled with zeroes to indicate that the path represented by that row does not exist. Based on the preceding, the CPM for chain 189 would be as shown in Table III.

Technically, neither the loop matrix nor the CPM as shown are true matrices since they are not rectangular arrays. However, the arrays as used in the computer program are true matrices as shown below.

To develop the CPM by computer, the loop matrix is required as the initial input. In most cases, developing the loop matrix is a simple matter but, for a certain class of chains such as Group 64D as shown in Appendix A, problems arise. These chains have paths which cross one another and, consequently, basic loops are not available. The loop matrices actually used for this class of chains (Appendix B) were

# TABLE I

LOOP MATRIX FOR CHAIN 189

ROW	1	301 301	209 402	406 406	207 209	208
ROW	3	402	210	305	406	
ROW	4	402	203	204	305	210

# TABLE II

COMBINATIONS OF LOOPS USED IN FORMING THE CPM

ROW #	Loop Combinations
1 2 3 4 5 6 7 8 9 10 11 12 13 14	Loop 1 Loop 2 Loop 3 Loop 4 Loops 1 and 2 Loops 1 and 2 Loops 1 and 3 Loops 2 and 3 Loops 2 and 4 Loops 3 and 4 Loops 1 and 2 and 3 Loops 1 and 2 and 4 Loops 1 and 2 and 4 Loops 1 and 3 and 4 Loops 2 and 3 and 4
15	Loops I and Z and 3 and 4

TABLE	III
-------	-----

CPM FOR CHAIN 189

ROW #				Sequ	uence	2		
1	301	209	406	207	208			
2	301	402	406	209				
3	402	210	305	406				
4	402	203	204	305	210			
5	301	402	406	207	208			
6	0	0	0	0	0			
7	0	0	0	0	0			
8	301	402	210	305	406	209		
9	0	0	0	0	0			
10	402	203	204	305	406			
11	301	402	210	305	406	207	208	
12	0	0	0	0	0			
13	0	0	0	0	0			
14	301	402	203	204	305	406	209	
15	301	402	203	204	305	406	207	208

TABLE IV

.

COMPUTER LOOP MATRIX F	OR CHAIN	189
------------------------	----------	-----

•	·····			······	<del>,</del>				<u></u>	;		·····	
	-301	209	406	207	-208	0	0	0	0	0	. 0	5	
	301	402	406	209	0	0	0	0	0	0	0	4	
	402	210	305	406	0	0	0	0	0	0	0	4	
	402	203	204	305	210	0	0	0	0	0	0	5	

arrived at by trial and error and no rules governing their development have become apparent.

The loop matrix is read into the computer in the form of a matrix and forms the first four (in the case of the ten-link chains) rows of the CPM. Sufficient columns are available in the matrix to contain the longest row which may occur (10 elements in the case of the ten-link chains) plus two. The two extra columns are required for use in certain operations performed in comparing rows. Any elements of the matrix which are not defined are filled with zeroes. As the loop matrix is read in, the number of non-zero elements in each row is counted and this number is stored in the last column of that row. Thus, the loop matrix (and the first four rows of the CPM for chain 189) would appear as shown in Table IV.

To form the remaining rows of the CPM, combinations of the first rows must be taken. In combining rows, the following rules are observed:

1. If an element is common to both rows and the elements on both sides of it are also common to both rows, that element will not appear in the combined row. For example, if rows 1 and 2 of Table IV are being combined, link 209 will not appear in the combined row since it is flanked by links 301 and 406 which are also common to both rows. Note that in row 2, 301 is considered to be adjacent to 209.

2. If row A is to be combined with row B, the path sequence will begin with  $A_1$  (the first element in row A) and will continue to element  $A_m$ , the first element encountered which is common to both rows. If rule 1 does not apply and if  $A_{m+1}$  is the same as  $B_{n-1}$  (where  $B_n$  is the element in row B which is equivalent to  $A_m$ ), the sequence will transfer

to row B at element  $B_n$  and will continue in row B until similar conditions permit a transfer back to row A to complete the path.

3. If no transfer can be made between rows, the resulting combined row will be filled with zeroes.

To illustrate the above procedure, the combining of rows 2 and 3 of the loop matrix for chain 189 (Table IV) will be performed step by step.

1. The first element of row 2 (301) is not common to both rows and will thus become the first element of the combined row.

2. The second element of row 2 (402) is the same as the first element of row 3. Rule 2 is applied to determine if a transfer can be made. Since the third element in row 2 (406) is the same as the last element in row 3 (which is considered as preceding the first element in row 3) the transfer can be made. The second element in the combined row thus becomes 402.

3. The sequence continues with elements 2 (210) and 3 (305) of row 3 since they are not common to both rows. At this point, the combined row is:

### 301 402 210 305.

4. The next element in row 3 (406) is common to both rows. Rule 2 is applied and, since the element following 406 in row 3 (402) is the same as the element preceding 406 in row 2, the sequence transfers back to row 2 and 406 becomes the fifth element of the combined row.

5. The sequence then continues in row 2 until the starting element is encountered. (When the sequence reaches the last element of a row, it starts over at the first element. Therefore, the first element of the row will be encountered after the last element of the row.)

6. The remainder of the combined row is filled with zeroes and the number of non-zero elements stored in the last column. The resulting row is:

301 402 210 305 406 209 0 0 0 0 0 6. This row agrees with row 8 of the CPM as shown in Table III.

Rows 5 through 10 of the CPM are similarly derived. Row 11 of the CPM represents the combination of rows 1, 2, and 3 of the loop matrix. This combination can most easily be obtained by combining row 1 of the CPM with row 8 of the CPM (the combination of rows 2 and 3) or by combining row 3 with row 5 (the combination of rows 1 and 2). Since one or more of the combined rows mentioned above may be zero, the computer tests the rows to be combined and uses the first pair which have nonzero entries in their first columns. The possible combinations to be tested are provided as input data for the program. Rows 11 through 15 of the CPM are generated in this manner. The row combinations used for the ten-link chains appear in Table V.

TABLE	V
-------	---

# ROW COMBINATIONS USED IN FORMING THE CPM FOR TEN-LINK CHAINS

 $(A_{i}) = (A_{i}) + (A_{$ 

ROW #	Combined Rows
5	1 and 2
6	1 and 3
7	1 and 4
8	2 and 3
9	2 and 4
10	3 and 4
11	1 and 8 or 5 and 3
12	4 and 5 or 7 and 2
13	10 and 1 or 3 and 7
14	8 and 4 or 2 and 10
15	5 and 10 or 6 and 9 or 7 and 8 or 12 and 3
	or 13 and 2 or 14 and 1

. ....

### CHAPTER IV

### DEVELOPMENT AND USE OF THE MECHANISM PATH MATRIX

### Theory

Once the CPM has been formed for a given chain, it represents the sequence of links encountered in traversing each possible path in the chain. The CPM could be used directly in comparing two chains but the procedure would involve checking each row in one matrix against each row in the other matrix and cycling one of the rows several times for each comparison. A method which will eliminate a great deal of the comparisons made in the above procedure is based on the premise that if a mechanism formed from one chain is identical to a mechanism formed from a second chain, the two chains are equivalent. Conversely, if a mechanism formed from one chain is not equivalent to any mechanism formed from the second chain, the chains are not equivalent. The problem thus becomes one of representing mechanisms and then, comparing two mechanisms.

As defined previously, a mechanism is created when one link of a kinematic chain is fixed with respect to ground. The configuration can be completely defined by defining all paths in the chain which <u>include</u> the fixed link. It will prove to be advantageous to use the fixed link as the starting element for each of the paths. A Mechanism Path Matrix (MPM) which describes the mechanism formed by fixing a specific link is developed by using only those rows in the CPM which contain the fixed

1 Q

link and by cycling each row so that the fixed link is the first element in the row. Table VI shows the CPM formed for chain 189 and Table VII, the MPM representing the mechanism formed by fixing link 1 of chain 189.

When comparing two paths, the sequence of link types encountered in traversing the paths is the factor which determines whether or not the paths are equivalent. Consequently, once the MPM is formed, the link numbers may be dropped leaving only the matrix of link types. Table VIII shows the MPM of Table VII reduced to a matrix of link types. The term Mechanism Path Matrix (MPM) will mean the matrix of link types in all following discussion.

To proceed further, the definition of equivalent paths must be established. Two paths are equivalent if and only if the sequence of elements encountered in traversing the first path in a given direction is the same as the sequence of elements encountered in traversing the second path in the same or opposite direction. That is, when comparing rows of two MPM's, the path represented by the sequence 2 3 2 3 4 is equivalent to the path represented by the sequence 2 4 3 2 3.

The equivalency of two MPM's, and therefore two mechanisms, will be established if each row in one path matrix has an equivalent row in the other path matrix, each row being used only once. Obviously, then, for two MPM's to be equivalent they must have the same number of rows and equivalent rows must contain the same number of non-zero elements. Only after these two conditions are satisfied is it necessary to compare sequences within two rows.

To illustrate the comparison of two MPM's, mechanisms 1 and 5 of chain 189 will be compared. The MPM for mechanism 5 is shown in Table

TABLE		۷	Τ
-------	--	---	---

301	209	406	207	208	0	0	0	0	0	0	5
~301	402	406	209	0	0	0	0	0	0	0	4
402	210	305	406	0	0	0	0	0	0	0	4
402	203	204	305	210	0	0	0	0	0	0	5
-301	402	406	207	208	0	0	0	0	0	0	5
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
<b>_301</b>	402	210	305	406	209	0	0	0	0	0	6
0	0	0	0	0	0	0	0	0	0	0	0
402	203	204	305	406	0	0	0	0	0	0	5
301	402	210	305	406	207	208	0	0	0	0	7
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
.301	402	203	204	305	406	209	0	0	0	0	7
301	402	203	204	305	406	207	208	0	0	0	8

COMPUTER CPM FOR CHAIN 189

### TABLE VII

MPM FOR MECHANISM 1 OF CHAIN 189

301	209	406	207	208	0	0	0	0	0	0	5	
301	402	406	209	0	0	0	0	0	0	0	4	
301	402	406	207	208	0	0	0	.0	0	0	5	
301	402	210	305	406	209	0	0	0	0	0	6	
301	402	210	305	406	207	208	0	0	0	0	7	
301	402	203	204	305	406	209	0	0	0	0	7	
301	402	203	204	305	406	207	208	0	0	0	8	

# TABLE VIII

MPM OF TABLE VII REDUCED TO MATRIX OF LINK TYPES

3	2	4	2	2	0	0	0	0	0	0	5	
3	4	4	2	0	0	0	0	0	0	0	4	
3	4	4	2	2	0	0	0	0	0	0	5	
3	4	2	3	4	2	0	0	0	0	0	6	
3	4	2	3	4	2	2	0	0	0	0	7	
3	4	2	2	3	4	2	0	0	0	0	7	
3	4	2	2	3	4	2	2	0	0	0	8	

IX and will be compared with the MPM in Table VIII.

First, the two matrices have the same number of rows thus justifying further comparisons. Row 1 of Table VIII contains 5 elements and, therefore, may be equivalent to either row 2 or row 4 of Table IX. It is found to be equivalent to row 2 of Table IX. Similarly, each row of Table VIII is found to be equivalent to a row in Table IX and vice versa. Therefore, the two mechanisms are equivalent.

As mentioned previously, two chains can be compared by comparing mechanisms formed from them. However, other initial comparisons will determine whether or not comparison of mechanisms is warranted. Obviously, the two chains must have the same number of links and the same (), (), number of each type of link. Also, the CPM's must have the same number (), of rows thus indicating that the two chains have the same number of paths. Once these conditions have been met, mechanisms from each chain should be chosen for comparison. The choice of mechanisms will often reduce the number of comparisons required.

For two mechanisms to be equivalent they must have the same type of fixed link. Thus, choosing the link type which appears least frequently in the chain will limit the number of mechanisms compared. For example, if two chains are to be compared and each chain has only one quarternary link, the only mechanisms which have to be compared are those formed from fixing the quarternary link.

# Computer Development of the MPM

Subroutine MECH (Appendix D) uses the CPM to develop the MPM for each mechanism derived from a given chain. To do this, it takes each link (in numerical order) and forms the MPM from rows in the CPM which

21 (1) Egyal Pr

(2)

· 3	•.	4	<b>4</b>	2	0	0	0	0	0	0	0	4
3		2	4	2	2	0	0	0	0	0	0	5
3		4	2	3	4	2	0	0	0	0	0	6
3		4	4	2	2	0	0	0	0	0	0	5
3		4	2	2	3	4	2	0	0	0	0	7
3		4	2	3	4	2	2	0	0	0	0	7
3		4	2	2	3	4	2	2	0	0	0	8

TABLE IX

MPM FOR MECHANISM 5 OF CHAIN 189

contain that link. As the MPM is formed, each row is cycled so that the fixed link appears as the first element. In addition, the number of rows in each MPM is determined and this number is stored in the (N-1)st column of the first row of the MPM. This permits a rapid determination of the number of rows when the MPM's are being compared.

### Computer Comparison of MPM's

Subroutine MECCOM (Appendix E) compares MPM's using the comparison procedure described earlier in this chapter as may be seen from the flow chart in Appendix E. An array of MPM's such as that developed by Subroutine MECH serves as the input to MECCOM. Each MPM is compared with all other MPM's and the results stored in array IEQUIV.

Two features are incorporated in Subroutine MECCOM which are not used in comparing the MPM's as they have been described thus far.

 Provisions are made for cycling one of two rows being compared. This feature is used in comparing modified MPM's described in following chapters.

2. Provisions are made for zeroing certain rows of array JSTORE and then compacting the non-zero rows. This feature is used in the development of the modified MPM's described in following chapters.

Computer Program for Determining the Ten-Link Mechanisms

The main program and subroutines (other than LOOP, MECH, and MECCOM) used in determining the ten link mechanisms are shown in Appendix F. One page of the output obtained for chains 1 through 25 is shown in Table X. In Table X, the columns represent link numbers one through ten and the rows represent the chain being described. An entry

# TABLE X

# SAMPLE OUTPUT FOR TEN-LINK MECHANISMS

			r		Li	INK I	NUMBE	E R				
CHAIN	11 ++	<b> </b> 	2	3 	4 	5	6	7	8	9 	10    +	UNIQUE MECHANISMS
1	1	 	1	   	3	1	1	3	3	 		3
2	1	   	1	•   	 	• } •		3	4	6	5	5
3		1			1			•				10
4	1				3	2	1			,		7
5										   		10
6	†1 			╊╼╼╼┥ ╏	╊╼┉═╶ ╏ ╵	₿		╏		   		10
7				╏	<b>;</b>	t			t	t 		10
8				2	1 1			6	5	 1	9	5
9	++			2	1	} 	5		5	5	7	4
10				2	1	<b>}</b> - 		•	5	6	7	5
11	+1			<b> </b>		<b> </b>	 			} 	}∦ 	10
12				}		 		 	 		+	10
13	† † 			┠╼═╼┥ ┨						 	┠━━━╂ ┃   ┃	10
14	++									┠╼╍╼╍ 	┠╼╼╼╂ ┃   ┃	10
15	++			╞╼┈╼┥ ╏						•		10
16	++				3	2	1			7	╉╾╼╼-╉ ┨╶──┨	6
17	++ 				2	   1			┣╼ <b>╌</b> ╼┤ ╏	} 	5	7
18	++						 		•	 	┨╴╴╴╸┨ ┃	10
19	++ 					┣╼╼╼┤ ┃	 		┟╼╶╾╼┥ ╏	} 	}+ 	10
20	++ 			} 	ł	╂╼╌──┤ ┟	}		┠═╼╾┥ ┨	} 	┠┈╼╼╊ ┃   ┃	l 10
21	++		}	2		}{ 	⊦   	┠╼╼╼┧ ┃	┠╼╺╾╸┤ ┨	} }	   9	+ 1 7
22	++ 			} 	┫╼╼╼╼╺ ┨	∦ 	 	}	┟╼┈┈┥ ┃	}   	}	10
23	++		1	}∤ 	}	   4	   3	3		4	3	3
24	++ 			}{ }	   1	   2	3	 	┠───┤ ┃	   8	+   7	+   5
25	++ 		┝╼╼╼┥	   1	┢═╌═╕ ╽	}   (\ _		}{   4	┠╼╼╼┥ ╏╴╷╴╷	} 	   8	   6
·						·	¥		y		ķ	

in any column indicates that the link designated by that column is equivalent to the link number entered in the column. For example, in row 1 representing chain 1, links 2, 5, and 6 are equivalent to link 1, links 4, 7, and 8 are equivalent to link 3, and link 10 is equivalent to link 9. A blank entry indicates that the mechanism formed by fixing the link corresponding to the column number is a unique mechanism. For chain 1 there are three blank columns indicating that fixing links 1, 3, or 9 will give three unique mechanisms. Fixing any other link will result in a mechanism equivalent to one of the above three. The total number of unique mechanisms for each chain is reflected in the last column.

The entire program for developing the ten-link mechanisms took 6 minutes and 50 seconds to run on an IBM 360 Model 50 computer. The total number of unique mechanisms obtained was 1836.
#### CHAPTER V

#### CHAINS WITH SPRINGS

In Chapter I it was indicated that the CPM is useful in converting basic kinematic chains into chains with various elements such as springs and cams. In this chapter, the development of chains with springs will be covered.

#### Theory

Two binary links connected by a revolute pair may be replaced by a spring as shown in Figure 7. In the CPM, this configuration would appear as two adjacent binary links in one or more rows. If a spring is designated by a type number, for example 11, then the two numbers representing the binary links can be replaced by the single number representing the spring.

From Figure 6 it can be seen that links 3 and 4 and links 7 and 8 of chain 189 can be replaced by springs. This information is also apparent in Table VI which shows the CPM for chain 189. In the CPM, binary links 3 and 4 are seen to be adjacent in rows 4, 10, 14, and 15 and binary links 7 and 8 are seen to be adjacent in rows 1, 5, 11, and 15. Thus, from the CPM it would appear that chain 189 can yield two chains with one spring, one with links 3 and 4 replaced by a spring and one with links 7 and 8 replaced by a spring. It is evident from Figure 6 that these two chains are equivalent due to the symmetry of chain



Figure 7. Two Binary Links Replaced by a Spring

189. However, a procedure must be devised to determine symmetry from the CPM.

If links 3 and 4 are replaced by a spring and an MPM formed as if the spring were a fixed link, the MPM would then represent all paths originating at the spring. Similarly, an MPM can be formed for the chain with links 7 and 8 replaced by a spring. The two MPM's would appear as shown in Tables XI and XII. The two MPM's are identical thus indicating that the two chains are equivalent.

To obtain a chain with two springs, both pairs of binary links are replaced with springs in the CPM. To compare the resulting chain with another chain containing two springs, the MPM's representing each spring could be compared. However, the same result may be obtained by using a modified MPM consisting of the combination of the individual MPM's for each spring. That is, the modified MPM will contain each row of the CPM which contains either spring or both springs. The modified MPM representing chain 189 is shown in Table XIII. Since the modified MPM does not represent the paths originating from a single element, the procedure for comparing two modified MPM's must be changed. Essentially, what is required is that, before deciding that two rows are not equivalent, a row containing two springs must be compared first with one spring as the first element and second, with the other spring as the first element. In Table XIII, only the last row contains two springs and the row would be the same regardless of which spring was used as the first element. However, for two rows, A and B, as shown in Table XIV, one of the rows must be cycled to obtain an equivalence. If row B is cycled as shown in the table, it becomes the same as row A read in reverse. Thus the two rows are equivalent.

14

MPM FOR	CHAIN	189	WITH	LINKS	3	AND 4	REPLA	CED 1	BY A	SPRING	ļ
11	3	2	4	0	0	0	0	0	0	0	4
11	3	4	4	0	0	0	0	0	0	0	4
11	3	4	2	3	4	0	0	0	0	0	6
11	3	4	2	2	3	4	0	0	0	0	7

TABLE XII

MPM FOR CHAIN 189 WITH LINKS 7 AND 8 REPLACED BY A SPRING

												ų
11	3	2	4	0	0	0	0	0	0	0	4	
11	3	4	4	0	0	0	0	0	0	0	4	
11	3	4	2	3	4	0	0	0	0	0	6	
11	3	4	2	2	3	4	0	0	0	0	7	

TABLE XIII

MODIFIED MPM FOR CHAIN 189 WITH TWO SPRINGS

11	3	2	4	0	0	0	0	0	0	0	4
11	3	2	4	0	0	0	0	0	0	0	4
11	3	4	4	0	0	0	0	0	0	0	4
11	3	4	4	0	0	0	0	0	0	0	4
11	3	4	2	3	4	0	0	0	0	0	6
11	3	4	2	3	4	0	0	0	0	0	6
11	3	4	11	3	4	0	0	0	0	0	6

TABLE	XIV

EQUIVALENCE OF ROWS WITH TWO SPRINGS

ROW	<b>A</b> .	11	3	4	11	4	3	3
ROW	В	11	4	3	11	3	3	4
ROW	B CYCLED	11	3	3	4 ]	L1	4	3

In the last chapter, it was mentioned that a certain portion of Subroutine MECCOM was used for cycling rows when required for the comparison of certain modified MPM's. That portion of MECCOM performs the cycling described above and permits comparison of modified MPM's representing any number of special elements. The number of times any row is cycled is limited to the number of identical special elements inserted in the chain. If a chain contained three springs, for example, a row would not be cycled more than three times.

Computer Program for Developing Chains With Springs

The computer program for developing chains with springs is shown in Appendix G along with a description of the variables used. The program performs the process of adding springs to a chain by proceeding as follows:

1. The loop matrix is read in and converted to the CPM. (Subroutine LOOP)

2. The CPM is examined and the pairs of links to be replaced by springs are assigned a number and stored in array LPAIR. (Subroutine SPRLOC)

The identification of the possible springs is printed out.
 (Subroutine ARRAY)

4. MPM's are developed for a single spring at each one of the locations determined in step 2 and are stored in array MX. (Subroutine SPRNG1)

5. The MPM's are compared and the equivalent chains identified. The results of the comparison are printed out and the unique chains, identified by the spring number assigned in step 2, are stored in array JSTORE. (Subroutine MECCOM)

6. A second spring is added to the unique chains identified in step 5 and the modified MPM's formed. (Subroutine SPRING2)

7. The modified MPM's are compared and the unique chains identified as in step 5. (Subroutine MECCOM)

8. Additional springs are added and unique chains determined until the chain can accept no more springs. (Subroutines SPRNG3 and SPRNG4)

A sample print out for chain 189 with springs is shown in Table XV. A summary of results obtained from the ten-link chains is shown in Table XVI.

#### TABLE XV

#### SAMPLE OUTPUT FOR CHAIN 189 WITH SPRINGS

CHAIN 189 \*\*\*\*\*\*\*\*\*

**IDENTIFICATION OF SPRINGS** 

SPRING	¥. − 1	I	1	ł	2	J
LINKL,	LINK2	1	3, 4	1	7,	8

CHAINS WITH ONE SPRING (IDENTIFIED BY THE SPRING NUMBER ABOVE)

EQUIVALENT CHAINS

1= 2

NUMBER OF UNIQUE CHAINS - 1

IDENTIFICATION OF CHAINS WITH TWO SPRINGS

CHAIN# | 1 | SPRINGS| 1,2 |

EQUIVALENT CHAINS

NCNE

NUMBER OF UNIQUE CHAINS - 1

NOTE: "LINK1, LINK-2" indicates the links replaced by the spring.

## TABLE XVI

## TEN-LINK CHAINS WITH SPRINGS

Total number of 10-link chains with 1 spring	-	234
Total number of 10-link chains with 2 springs	-	83
Total number of 10-link chains with 3 springs	-	12
Total number of 10-link chains with 4 springs	<b>-</b> *	1

#### CHAPTER VI

#### CHAINS WITH PULLEYS AND BELTS

#### Theory

A ternary link with two binary links connected to it can be converted to a pulley and belt as shown in Figure 8. In the CPM, such an arrangement of links would appear as a ternary link with a binary link on each side of it. If the ternary link is replaced by some type number designating a pulley, say 11 again, and the binary links are replaced by some type number representing a belt, say 12, the CPM will then represent the chain with one pulley. An MPM can be formed using the pulley as the starting element for each path and will serve as a means of comparing one chain with a pulley to a second chain with a pulley.

As an example of the above, consider link 1 of chain 189 (Figure 6). Since it is a ternary link connected to two binary links, it can be converted to a pulley and belt. This fact is reflected in row 1 of the CPM for chain 189 as shown in Table VI where 301 appears between 208 and 209. To develop the MPM with link 1 converted to a pulley, 301 must be converted to 11 everywhere it appears in the CPM. Similarly, links 208 and 209 must be converted to 12 wherever they appear The MPM with the above conversions completed is shown in Table XVII.

Some kinematic chains may contain several pulleys just as some may contain several springs. However, some rules must be established to

21,



. .

Figure 8. Ternary Link and Two Binary Links Converted to a Pulley and Belt

### TABLE XVII

#### MPM FOR CHAIN 189 WITH LINK 1 CONVERTED TO A PULLEY

									<del></del>		
11	12	4	2	12	0	0	0	0	0	0	5
11	4	4	12	0	0	0	0	0	0	0	4
11	4	4	2	12	0	0	0	0	Ō	0	5
11	4	2	3	4	12	0	0	0	0	0	6
11	4	2	3	4	2	12	0	0	0	0	7
11	4	2	2	3	4	12	0	0	0	0	7
11	4	2	2	3	4	2	12	0	0	0	8

AND LINKS 8 AND 9 CONVERTED TO A BELT

prevent the chain from becoming a structure. The rules established in converting the ten-link chains to chains with pulleys are given below and illustrated in the referenced figures.

1. A ternary link joined to three binary links may be converted to three different pulley/belt combinations. However, no two of these can exist simultaneously. (Figure 9)

Two pulleys may share a common belt, but two binary links connected by a revolute pair may not be converted to a single belt.
 (Figure 10)

## Computer Program for Converting Basic Kinematic Chains to Chains with Pulleys and Belts

Essentially, the same procedure is followed in developing chains with pulleys as was followed in the preceding chapter where springs were used. The possible locations for pulleys are determined and then, the pulleys are added one at a time to establish the chains. Subroutine PULLOC, which determines the pulley/belt locations, and Subroutine PUL3, which develops the modified MPM for chains with three pulleys, are shown in Appendix H. Subroutines for adding more or less pulleys follow the same procedures as PUL3. The entire program for adding pulleys is the same as the program for adding springs (Appendix G) except that Subroutine PULLOC would replace Subroutine SPRLOC and Subroutines PUL1, PUL2, PUL3, and PUL4 would replace Subroutines SPRNG1, SPRNG2, SPRNG3, and SPRNG4. All other subroutines are unchanged except for format statements. The computer print out for chain 189 with pulleys is shown in Table XVIII. A summary of the results obtained for the ten-link chains with pulleys appears in Table XIX.



## Figure 9. Ternary Link with 3 Binary Links Converted to Pulleys









.

Figure 10. Ternary Links Sharing a Binary Link Converted to Pulleys and Belt

#### TABLE XVIII

## SAMPLE OUTPUT FOR CHAIN 189 WITH PULLEYS

CHAIN 189 \*\*\*\*\*\*\*\*\*

IDENTIFICATION OF PULLEYS

 PULLEY#
 1
 1
 2
 1

 PULLEY(BELT, BELT)
 1(8,9)
 5(4,10)

CHAINS WITH ONE PULLEY (IDENTIFIED BY PULLEY NUMBER ABOVE)

EQUIVALENT CHAINS

1 = 2

NUMBER OF UNIQUE CHAINS - 1

IDENTIFICATION OF CHAINS WITH TWO PULLEYS

۰,

CHAIN# 1 1 PULLEYSI 1,2

EQUIVALENT CHAINS

NONE

NUMBER OF UNIQUE CHAINS - 1

.

NOTE: "PULLEY (BELT, BELT)" refers to the link replaced by a pulley and the two links replaced by a belt.

## TABLE XIX

## TEN-LINK CHAINS WITH PULLEYS

Total number of 10-1ink chains with 1 pulle	ey - 358
Total number of 10-link chains with 2 pulle	eys - 240
Total number of 10-link chains with 3 pulle	eys - 58
Total number of 10-link chains with 4 pull	eys - 7

`.

#### CHAPTER VII

#### CHAINS WITH CAM PAIRS

A binary link and two revolute pairs may be converted to a cam pair as shown in Figure 11. In a basic kinematic chain, all links are joined by revolute pairs, therefore, each binary link may be replaced by a cam pair. If a cam pair is represented by a type number, say 11, the CPM may be converted to an MPM representing a chain with one cam pair simply by replacing a binary link with the number 11. Additional cam pairs may be added and modified MPM's formed subject to the restriction that two adjacent binary links cannot be simultaneously converted to cam pairs. This means that, in Figure 11, links 2 and 3 cannot both be converted to cam pairs.

The computer program for developing chains with cam pairs follows the same pattern as those for springs and for pulleys and belts. Subroutine CAMLOC, which locates the cam positions, and Subroutine CAM3, which corresponds to SPRNG3 and PUL3 in previous chapters, are shown in Appendix I. The computer print out for chain 189 with cam pairs is shown in Table XX. A summary of the results obtained for the ten-link chains with cam pairs is shown in Table XXI.



Figure 11. Binary Link Converted to a Cam Pair

.

#### TABLE XX

#### SAMPLE OUTPUT FOR CHAIN 189 WITH CAM PAIRS

CHAIN 185

IDENTIFICATION OF CAM PAIRS

CAMN	ł	1	I	2	1	3	1	4	1	5	1	6	I
LINK	1.	3		4	 	7	 	8	]	9	1	10	1

CHAINS WITH ONE CAM PAIR (IDENTIFIED BY THE CAP NUMBER ABOVE)

.

.

ECUIVALENT CHAINS

1= 3 2= 4 5= 6

NUMBER OF UNIQUE CHAINS - 3

IDENTIFICATION OF CHAINS WITH TWO CAM PAIRS

CHAINN	I	1	1 <sup>°</sup>	2	1	3	1	4	1	5	1
CAMS	1	1,3		1.4	1	1,5	1	1,6		2,3	1
CHAINS	I	6	ł	7	ļ	8	I	9	I		
CAPS	l	2.4	1	2,5	ļ	2,6	1	5,6			·

EQUIVALENT CHAINS

2≖ 5

NUMBER OF UNIQUE CHAINS - 8

IDENTIFICATION OF CHAINS WITH THREE CAM PAIRS

CHAINS	ł	1	I	2	i	3	1	4 1	5	I
CAMS	1	1,3,5	1	1,3,6	]	1,4,5	ļ	1,4,6	1,5,6	- , 
CHAINS	ł	6	i	7	i	8	1			
CAMS	1	2,4,5		2,4,6	1	2,5,6				

TABLE XX (Continued)

CHAIN 189 (CONTINUED)

EQUIVALENT CHAINS

1= 2 6= 7

NUMBER OF UNIQUE CHAINS - 6

# ICENTIFICATION OF CHAINS WITH FOUR CAM PAIRS

CHAIN#	1	1	1	2	1 .	3	t i
CAMS	1	1,3,5,6	1	1,4,5,6	ł	2,4,5,6	1

EQUIVALENT CHAINS

NCNE

NUMBER OF UNIQUE CHAINS - 3

IDENTIFICATION OF CHAINS WITH FIVE CAM PAIRS

NC POSSIBLE CHAINS

;

## TABLE XXI

## TEN-LINK CHAINS WITH CAM PAIRS

Total number of	E 10-link	chains	with	1	cam pai	r -	913
Total number of	E 10 <b>-link</b>	chains	with	2	cam pai	rs -	1661
Total number of	f 10-link	chains	with	3	c <b>a</b> m pai	rs -	1415
Total number of	E 10-link	chains	with	4	cam pai	rs -	624
Total number of	E 10-link	chains	with	5	cam pai	rs -	121
Total number of	E 10-link	chains	with	6	cam pai	rs -	10

#### CHAPTER VIII

#### CHAINS WITH PRISM PAIRS

A prism pair may replace a revolute pair in a kinematic chain thus permitting linear motion rather than rotary motion. If a prism pair is represented by a specific type number, say 11 once more, converting the CPM to represent a chain with one prism pair becomes a simple matter of inserting the 11 between the two links to be joined by the prism pair. The MPM is then formed from each row of the CPM containing a prism pair. Chains with two prism pairs may also be formed in the same manner and a modified MPM used for comparison of those chains just as in previous chapters.

If more than two prism pairs are added to a chain, care must be taken to insure that the mobility of the chain is retained. If three prism pairs are inserted in a circuit or path with only four links, the mobility of the chain is no longer retained. Similarly, four prism pairs in a circuit consisting of five links will result in that circuit becoming underconstrained although constrained motion is achieved with three prism pairs (with non-parallel axes). Since every path represented in the CPM actually represents a complete circuit, the requirement for mobility reduces to the rule that no path may contain more than n-2 prism pairs where n is the number of links in the path. Thus, to insure that mobility of a chain is maintained when adding more than two prism pairs, each row of the CPM must be checked to determine

if it will contain more than n-2 prism pairs.

The computer program for developing chains with prism pairs is similar to those developed in previous chapters. Subroutine JOINT locates all joints in the chain and identifies them according to the links incident to that joint. For example, the joint connecting links 2 and 3 in a chain would be identified by the notation 2-3.

Subroutine PRISM2 demonstrates the logic used in developing chains with multiple prism pairs. As was the case with double joints, the number of chains with more than two prism pairs which may be developed from a single basic kinematic chain becomes very large and therefore, the program used for the ten-link chains was limited to two prism pairs. For this reason, the test for mobility was not included in the program. Subroutines JOINT and PRISM2 are shown in Appendix J. The computer print out for chain 189 with prism pairs appears in Table XXII. A summary of the results obtained for all ten-link chains with prism pairs is shown in Table XXIII.

#### TABLE XXII

## SAMPLE OUTPUT FOR CHAIN 189 WITH PRISM PAIRS

.

A second second second second

#### CHAIN 189 \*\*\*\*\*\*\*

IDENTIFICATION OF JOINTS

JOINT#		١	1	1	2	1	3	1	4	1	5	I	6	1	7	1
LINK1,	LINK2	1	1, 2	1	1, 8	1	1, 9	1	2,3	1	2,6	1	2,10	1	3,4	Ī
JOINT#		ł	8	1	9	1	10	۱	11	I	12	1	13	ļ		
LINK1,	LINK2	1	4, 5	1	5,6	1	5,10	1	6, 7	1	6, 9	1	7,8		•	

# CHAINS WITH ONE PRISM PAIR (IDENTIFIED BY THE JOINT NUMBER ABOVE)

EQUIVALENT CHAINS

<u>l</u> ==	9
2=	8
3=1	0
4=]	1
6=1	. 2
7=1	13

NUMBER OF UNIQUE CHAINS - 7

# IDENTIFICATION OF CHAINS WITH TWO PRISM PAIRS

•

CHAIN#	1	1	2	ł	3	1	4	1.	5	1997 <b>1</b>
PULLEYS	1,2		1,3	1	1,4	· 1	1,5		1,6	
CHAIN#	6	ľ	7.	1	8	<b>.</b> I	9		10	ан 1
PULLEYS	1,7		1,8	1	1,9	 	1,10		1,11	 
CHAIN#	11	1	12	I	13	1	14	1	15	1
PULLEYS	1,12	1	1,13	1	2,3	1	2,4	, <b>I</b>	2,5	, I
CHAIN#	16	I	17	1	18	1	19	- 	20	с П. І
PULLEYS	2,6	<u> </u>	2,7		2,8	· 1	2,9	2 <b>1</b> -	,2,10	

NOTE: The word "PULLEYS" should read "JOINTS" in this table.

. .

i

CHAIN#	21	1	22	1	23		24	1	25	
PULLEYS	2,11	1	2,12	1	2,13	1	3,4	1	3,5	-
CHAIN#	26	1	27	ł	28	ŧ	29	ł	30	
PULLEYS	3,6		3,7	1	3,8		3,9		3,10	
CHAIN#	31	I	32	I	33	I	34	I	35	
PULLEYS	3,11		3,12	1	3,13	 	4,5		4,6	-
CHAIN#	36	I	37	I	38	1	39	ł	40	
PULLEYS	4,7	1	4,8	. 1	4,9	 I	4,10		4,11	
CHAIN#	41	ł	42	I	43	I	44	I	45	
PULLEYS	4,12		4,13		5,6	1	5,7		5,8	
CHAIN#	46	I	47	I	48	i	49	1	50	
PULLEYSI	5,9	l	5,10	I	5,11	1	5,12	I	5,13	
CHAIN#	51	l	52	1	53	1	54	1	55	
PULLEYS	6,7		6,8	1	6,9	1	6,10	1	6,11	
CHAIN#	56	.1	57	1	58	-	59	1	60	
PULLEYS	6,12	1	6,13	1	7,8	1	7,9	I	7,10	
CHAIN#	61	- <b>I</b>	62	1	63	1				
PULLEYS	7,11	1	7,12		7,13	1.				

7=19 9=29 10=38 11=53 12=59 15=45 20=28 21=37

. .

## TABLE XXII (Continued)

.

NUMBER OF UNIQUE CHAINS -43

## TABLE XXIII

## TEN-LINK CHAINS WITH PRISM PAIRS

Total	number	of	10-link	chains	with	1	prism pair	-		2312
Total	number	of	10-link	chains	with	2	prism pairs	-	1	2969

#### CHAPTER IX

#### CHAINS WITH DOUBLE JOINTS

#### Theory

Ternary and higher links may be converted to links with double joints by reducing a side between two revolute pairs to zero. Some examples of this are shown in Figure 12. When a double joint is added, care must be taken to insure that the chain retains a single degree of freedom. Rules ro prevent all or part of a chain from becoming a structure are as follows:

1. No loop may be reduced to only three links. (Figure 13)

2. No two paths, each containing only four links, may have three links in common. In Figure 14, this rule is demonstrated by loops 2 and 3. Initially, loop 2 contains 5 links, but, when a double joint is added such that links 7 and 5 are joined, loop 2 is reduced to four links. The portion of the chain composed of loops 2 and 3 becomes a structure which cannot move. (Although the figure shows an eight-link chain, this rule was derived from, and applied to, the ten-link chains).

In the discussion that follows, double joints will be identified by a set of numbers of the form l(2,3). The first number represents the link which has one side reduced to zero and will be referred to as the base link. The two numbers in parentheses represent the two links joined to the base link at the double joint. For example, the double joint shown in Figure 12 (a) would be referred to as 4(3,5).



Figure 12. Addition of Double Joint to Basic Kinematic Chain





Figure 13. Demonstration of Rule 1 for Double Joints



Figure 14. Demonstration of Rule 2 for Double Joints

Since double joints can be formed with any ternary or higher link, all possible double joints in a chain can be easily located by use of the CPM. In a row of the CPM, any link other than a binary link may be a base link, in which case, the two links adjacent to it become the other links incident at the double joint. For example, possible double joints found from the first two rows of the CPM for chain 189 (Table VI) are 1(8,9), 6(7,9), 1(2,9), 2(1,6), and 6(2,9). Some of these cannot actually be used due to rule 1 or 2 above. A double joint at position 1(2,9) would reduce row 2 of the CPM to a row with only three links (link 1 would effectively be removed from that path) thus violating rule 1. Positions 2(1,6) and 6(2,9) cannot be used for the same reason.

As in previous chapters, if a double joint is represented by a specific type number such as 11, an MPM may be developed which represents the chain with a double joint. In forming the MPM, the base link is replaced by the number 11 in those rows where it is flanked by the incident links. In other rows where the base link and only one of the incident links appear, the number 11 is inserted between the base link and the incident link. Table XXIV shows the MPM (with complete link identification numbers) for chain 189 with a double joint at position 1(8,9). The double joint is represented by 1100 to conform with the convention that the type number appears in the hundreds and thousands places.

Once the MPM has been established, chains with one double joint may be compared just as chains with other special elements. Additional double joints may also be added and modified MPM's formed. However, for each double joint added, the rules for mobility must be reapplied.

## MPM FOR CHAIN 189 WITH DOUBLE JOINT AT POSITION 1(8,9)

1100	209	406	207	208	0	0	0	0	0	0	5
1100	301	402	406	209	0	0	0	0	0	0	5
1100	301	402	406	207	208	0	0	0	0	0	6
1100	301	402	210	305	406	209	0	0	0	0	7
1100	301	402	210	305	406	207	208	0	0	0	8
1100	301	402	203	204	305	406	209	0	0	0	8
1100	301	402	203	204	305	406	207	208	0	0	9

In addition, a third restriction that arises when more than one double joint is used is that no two double joints may have two elements in common. That is, they may not share two links. For the chain shown in Figure 21 (b), joints 4(3,7) and 4(5,8) could exist simultaneously but 4(3,7) and 4(3,5) could not. The combination of position 4(3,7) and 4(3,8) would, in effect, create a triple joint which is not considered here.

#### Computer Program for Developing Chains With Double Joints

The basic program for developing chains with double joints follows the same pattern as previous programs. Subroutine DBLOC, which locates the positions of double joints, and subroutine DBL2, which constructs the modified MPM for chains with two double joints, appear in Appendix K. Subroutine DBL2 also tests for mobility by calling Subroutines LNKCNT and MOBCK. The computer print out for chain 189 with double joints appears in Table XXV.

The ten-link chains with more than two double joints were not developed due to the time required to perform the computations. In testing the program, it was found that a number of the ten-link chains would each produce in excess of 130 chains with three double joints. The computer time required to compare these chains runs into hours and was therefore not attempted. A summary of the results obtained for the ten-link chains with double joints appears in Table XXVI.

#### TABLE XXV

#### SAMPLE OUTPUT FOR CHAIN 189 WITH DOUBLE JOINTS

CHAIN 189

**IDENTIFICATION OF DOUBLE JOINTS** 

JOINT#	I		1	1		2	ł		3	ł		4	ł
BASE(LINK1,LINK2	)	1(	8,	9)	11	8,	211	21	1,	10)	2(	1,	3)
JGINT#	ł		5	i		6	1		7	I		. 8	1
BASE(LINK1,LINK2	)	21	6,	3)	21	10,	3)	51	10,	4)	5(	6,	4)
JOINT#	I		9	1		10	ł		11	j		12	<u></u> 1
BASE(LINK1,LINK2	11	61	9,	7)	61	9,	5)	61	7,	211	6(	7,	511

CHAINS WITH ONE DOUBLE JOINT (IDENTIFIED BY THE JOINT NUMBER ABOVE)

1

EQUIVALENT CHAINS

1= 7 2= 8 3=10 4=12 5=11 6= 9

NUMBER OF UNIQUE CHAINS - 6

IDENTIFICATION OF CHAINS WITH TWO DOUBLE JGINTS

CHAIN#	ł	L	1	2	i	3	1.	4	I	5	<b>.</b>
JOINTS	1	1,3	I	1,4	ł	1,5	ł	1,6	1	1,7	1
CHAIN#	I.	6		· 7	1	8	1	9	1	10	1
JOINTS	1	1,8	- 1	1,10	<u>(</u> 1 \	1,11	1	1,12	1	2,5	1
CHAIN#	i	11	ł	12	i	13	ļ	14	2 <b>I</b> 5	15	I,
JOINTS	1	2,6	i	2,7	1	2,8	ļ	2,9	1	2,10	1

12

.

	3,7 23 3,12 28	1	3,8 24 4,7 29	1 	3,9 25 4,8	
	23 3,12 28	   	24 4,7 29		25 4,8	
	<b>3,</b> 12 28		4 <b>,</b> 7 29	1	4,8	
	28	· 1	29			
0 1				1	30	
	4,11	1	4,12	1	5,7	
2 1	33	1	34	1	35	
.0	5,12	1	6,8	1	6,9	
· i	38					
1	6,12					
	i 1 1	38   6,12	i 38   1   6,12	i 38   1   6,12	i 38   1   6,12	i 38   1   6,12

.

ł

NUMBER OF UNIQUE CHAINS -25

.

## TABLE XXVI

## TEN-LINK CHAINS WITH DOUBLE JOINTS

Total	number	of	10-link	chains	with	1	double	joint	-	2039
Total	number	of	10-link	chains	with	2	double	joints	-	7575

#### CHAPTER X

#### COMPARISON OF MECHANISMS DERIVED FROM

#### CHAINS WITH SPECIAL ELEMENTS

Mechanisms derived from chains with special elements may be compared by use of the MPM. To demonstrate this and to further clarify the derivation of the CPM and MPM, three examples will be used. In these examples, the various special elements will be represented by type numbers as follows:

Pulley	-	12
Belt	-	13
Spring	-	14
Prism Pair	-	15
Double Joint	-	16
Cam Pair	-	17

#### Example 1

As the first example, chain number 4 from Appendix A will be used. Special elements will be added as follows:

1. Links 3 and 4 replaced by spring

2. Prism pair added at joint 8-9

3. Link 7 (and associated revolute pairs) replaced by a cam

pair.

The resulting chain would appear as shown in Figure 15.

The loop matrix for the basic chain appears in Table XXVII. To obtain the CPM for the modified chain, the CPM for the basic chain may be formed and the special elements added. An alternate approach would




### TABLE XXVII

### LOOP MATRIX FOR CHAIN 4

 										·····		<del></del>
302	203	204	305	310	0	0	0	0	0	0	5	
305	306	308	209	310	0	0	0	0	0	0	5	
306	207	301	308	0	0	0	0	0	0	0	4	
301	302	310	209	308	0	0	0	0	0	0	5	

## TABLE XXVIII

LOOP MATRIX FOR CHAIN 4 WITH SPECIAL ELEMENTS

302	1400	305	310	0	0	0	0	0	0	0	4
305	306	308	1500	209	310	0	0	0	0	0	6
306	1700	301	308	0	0	0	0	0	0	0	4
301	302	310	209	1500	308	0	0	0	0	0	6

### TABLE XXIX

CPM FOR CHAIN 4 WITH SPECIAL ELEMENTS

302	1400	305	310	0	0	0	0	0	0	0	4
305	306	308	1500	209	310	0	0	0	0	0	6
306	1700	301	308	0	0	0	0	0	0	0	4
301	302	310	209	1500	308	0	0	0	0	0	6
302	1400	305	306	308	1500	209	310	0	0	0	8
0	0	0	0	0	0	0	0	0	0	0	0
302	1400	305	310	209	1500	308	301	0	0	0	8
305	306	1700	301	308	1500	209	310	0	0	0	8
305	306	308	301	302	310	0	0	0	0	0	6
306	1700	301	302	310	209	1500	308	0	0	0	8
302	1400	305	306	1700	301	308	1500	209	310	0	10
302	1400	305	306	308	301	0	0	0	0	0	6
302	1400	305	310	209	1500	308	306	1700	301	0	10
305	306	1700	301	302	310	0	0	0	0	Q	6
302	1400	305	306	1700	301	0	0	0	0	0	6

be to form the loop matrix for the modified chain and then use it to form the CPM. The latter approach will be used here.

The loop matrix for the modified chain appears in Table XXVIII. The special elements have been added according to the rules established in previous chapters. This loop matrix is then used to form the CPM (Table XXIX). The next step is to form the MPM's for all mechanisms which can be derived from the modified chain.

The basic chain could have been converted to 10 mechanisms (1 mechanism per link), however, in the modified chain, links 3, 4, and 7 have been replaced by elements which cannot be fixed. Consequently, mechanisms 3, 4, and 7 cannot exist. This fact is obvious from the CPM since link numbers 3, 4, and 7 do not appear. Forming the MPM's for the remaining links yields the results shown in Table XXX.

To compare the mechanisms, each MPM must be compared with all other MPM's. Starting with MPM's 1 and 2, it is seen that each has 11 rows and therefore, further comparison is justified. Comparing row 1 of MPM #1 with row 1 of MPM #2 (both have 4 elements) shows that the two mechanisms are not equivalent since these rows are not equivalent and no other rows with 4 elements exist in CPM #2. Checking all other matrices shows that only MPM #6 has a 4 element row that is equivalent to row 1 of MPM #1. Further comparisons show the following equivalent rows:

MPM #	1	MPM #6
2	=	1
3	=	3
4	=	6
5	=	5
6	=	4
7	=	9
8	=	8
9	=	7
10	=	10
11	=	11

## TABLE XXX

### MPM'S FOR CHAIN 4 WITH SPECIAL ELEMENTS

NOM I	¥ 1												MPM	6		-								
	3 -	3	3	17	0	0	0	0	0	0	0	4		3 3	15	2	3	3	0	С	0	0	0	6
-	ŝ	3	3	2	15	3	0	0	е	0	0	-6	1	8 17	3	.3	0	е	0	0	G	0	0	4
	,	3	14	3	3	2	15	3	С	C	0	8	1	3 3	15	2	3	3	14	3	0	0	0	8
	,	3	15	2	3	3	3	17	0	0	0	8		3 17	3	3	15	2	3	3	0	0	0	8
	á	3	3	3	3	3	0	0	0	Ċ.	0	6		3 3	3	3	3	3	0	0	0	0	0	6
-	2	2	á	2	15	3	3	17	C	c	0	8		3 17	3	3	3	2	15	3	C	0	0	8
	á	ž	15	2	3	3	14	3	3	17	C	10		3 17	3	3	15	2	3	3	14	3	C	10
	2	2	14	3	3	3	0	C	e	0	с	6		3 3	3	3	14	3	0	Ċ	C	Ċ	Ó	6
	5	2	14	ă	Ä	2	15	3	3	17	С	10		3 17	3	3	14	3	3	2	15	3	0	10
-	2	2	2	2	3	17	0	0	0	С	С	6		3 17	3	3	. 3	3	0	0	0	0	0	6
	2	2	14	ž	3	17	Ō	0	с	0	0	6	-	3 17	3	3	14	3	0	G	0	0	0	6
	, 																							
MDM	¥ 7												MPN	8										
	~ <b>-</b>	14	3	3	0	0	0	0	е	0	0	4		3 15	2	3	3	3	0	· 0	C	0	0	6
	a a	Ĩ	2	15	3	3	0	0	C	Û	G	6	5	3 3	17	3	0	С	0	0	С	0	0	4
:	2	14	3	- 3	3	15	2	3	0	0	0	8		3 3	3	3	2	15	0	0	C	С	0	6
	a a	14	จั	3	2	15	3	3	C	Û	0	8	1	3 15	2	3	3	14	3	3	0	0	0	8
	à	3	ž	à	3	3	0	0	0	0	0	6		3 3	3	14	3	3	2	15	0	0	0	8
	a l	á	2	15	3	3	17	3	0	Û	0	8		3 15	2	3	3	3	17	3	0	0	0	8
	- -	14	3	3	17	3	3	15	2	3	0	10	-	3 - 3	3	3	3	3	0	0	G	0	0	6
	á	14	3	3	3	3	0	O	e	C	0	6		3 3	17	3	3	3	2	15	C	0	0	8
	2	14	3	3	2	15	3	3	17	3	0	10		3 15	2	3.	3	14	3	3	17	3	0	10
	ž	<b>1</b>	ž	3	17	3	0	0	0	0	0	6		3 3	3	14	3	3	0	0	0	0	0	6
	à	14	3	3	17	3	0	0	С.	0	0	6		3 3	17	3	3	14	3	3	Z	15	0	10
																						• • • • •		
MPM	# 5												MPM (	¥ 9										
	3	3	3	14	0	0	0	0	0	0	0	4		2 3	3	3	3	15	0	0	0	0	0	6
	3	3	.3	15	2	3	0	0	0	0	0	6		2 15	3	3	3	3	0	0	0	0	0	6
	3	3	3	15	2	3	3	14	0	0	0	8		2 3	3	14	3	3	3	15	0	0	0	8
	3	3	2	15	3	3	3	14	0	0	.0	8		2 15	3	3	3	14	3	3	0	0	0	8
	3	3	17	3	3	15	2	3	C	0	0	8		2 3	3	3	17	3	3	15	0	0	0	8
	3	3	3	3	3	3	0	0	Ĉ	0	0	6	-	2 15	3	3	17	3	3	3	C	0	0	8
	3	3	17	3	3	15	2	3	3	14	0	10		2 3	3	14	3	3	17	3	3	15	0	10
	3	3	3	3	3	14	0	0	0	С	0	6		2 15	3	3	17	3	3	14	3	3	0	10
	3	3	2	15	3	3	17	3	3	14	0	10								• • • • •		• • • • •	••••	
	3	3	17	3	3	3	0	0	e	0	0	6	MPM	£10				_		_				
	3	3	17	3	3	14	e	e	Ċ.	0	0	6		3 3,	14	3	0	0	G	D	0	0	0	4
									• • • • •	• • • • • •	• • • • •	• • • •		3 3	3	3	15	2	0	0	0	0	0	6
								•						3 2	15	3	3	3	0	0	0	0	0	6
														3 3	14	3	3	3	15	2	Ģ	0	0	8
														3 Z	15	3	3	3	14	3	0	0	0	8
														33	3	17	3	3	15	2	0	G	0	8
														3 3	3	3	3	3	0	0	0	0	0	6
														32	15	3	3	17	د	د		0	U	8
														3 3	14	3	3	17	د	د	15	2	0	10
														32	15	3	3	17	3	3	14	3	0	10
														3 3	3	17	3	3	Q	U	Ð	U	U	6

.........

64

. . . . . . . . . . . . .

Since each row in MPM #1 is equivalent to a row in MPM #6, the two mechanisms, 1 and 6, are equivalent. Similarly, comparison of MPM's 2 and 5 shows that they are equivalent. No other equivalencies exist, hence, five unique mechanisms (1, 2, 8, 9, and 10) may be formed from chain 4 with the special elements as shown in Figure 15.

## Example 2

The second example is based on chain number 48 from Appendix A. Special elements will be added as follows:

1. Link 4 replaced by a pulley

2. Links 3 and 8 replaced by a belt

3. Prism pairs added at joints 5-6 and 1-5

The resulting chain will appear as shown in Figure 16.

Since chain 48 is one of the chains with crossed links, the loops required to form the loop matrix are not easily identified. Therefore, the loop matrix for chain 48 as shown in Appendix B is used to form the loop matrix with special elements added (Table XXXI). Note that the special elements have been assigned "link numbers." This has been done in order to differentiate between elements of the same type so that incorrect row transfers will not take place when forming the CPM.

The CPM (Table XXXII) is formed by combining various rows of the loop matrix as explained in Chapter III. The process of combining rows is especially useful in this case since visual determination of the required paths is quite difficult.

The MPM's for all mechanisms which may be formed from chain 48 with the special elements added are shown in Table XXXIII. Note that MPM's 3, 4, and 8 do not appear since the corresponding links have been



Figure 16. Chain 48 with Special Elements Added

## TABLE XXXI

3

٠.

·									······		
301	1514	305	1515	306	210	302	0	0	0	0	7
301	302	1312	1211	305	1514	0	0	0	0	0	6
1211	1313	307	306	1515	305	0	0	0	0	0	6
301	1514	305	1515	306	307	209	0	0	0	0	7

LOOP MATRIX FOR CHAIN 48 WITH SPECIAL ELEMENTS

### TABLE XXXII

301	1514	305	1515	306	210	302	0	0	0	0	0	0	7
301	302	1312	1211	305	1514	0	0	0	0	0	0	0	6
1211	1313	307	306	1515	305	0	0	0	0	0	0	0	6
301	1514	305	1515	306	307	209	0	0	0	0	0	0	7
305	1515	306	210	. 302	1312	1211	0	0	0	0	0	0	7
301	1514	305	1211	1313	307	306	210	302	0	0	0	0	9
0	0	0	0	0	0	0	0	0	0	0	0	0	0
301	302	1312	1211	1313	307	306	1515	305	1514	0	0	0	10
301	302	1312	1211	305	1515	306	307	209	0	0	0	0	9
1211	1313	307	209	301	1514	305	0	0	0	0	0	0	7
306	210	302	1312	1211	1313	307	0	0	0	0	0	0	7
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
301	302	1312	1211	1313	307	209	0	0	0	0	0	0	7
305	1515	306	210	302	1312	1211	1313	307	209	301	1514	0	12

CPM FOR CHAIN 48 WITH SPECIAL ELEMENTS

# TABLE XXXIII

# MPM'S FOR CHAIN 48 WITH SPECIAL ELEMENTS

.

мРм	# ] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	15 3 15 3 15 3 15 3 15	3 13 3 13 13 13 3 13 3	15 12 15 12 12 12 12 12 12 12 12	3 3 13 13 13 13 13 3	2 15 3 3 15 3 3 2	3 0 2 3 3 3 2 2 3	C 0 2 15 3 0 C 13	C 0 3 3 2 0 12	C 0 0 15 0 0 13	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 2	0 0 0 0 0 0 0 0	7 6 7 9 10 9 7 7 12	мФм -	# 7 3 3 3 3 3 3 3 3 3 3 3 3 3 3	3 2 3 2 2 3 2 2 3 2 2 2	15 2 15 3 2 3 3 3	3 3 3 15 3 15 3 15	12 3 15 13 3 13 13	13 15 15 3 12 12 12 12 12 12 15	0 3 3 13 13 13 3	0 0 12 13 15 0 0 2	0 0 13 12 3 0 0 0 0 3	0 0 13 0 0 0 13	0 0 0 0 0 0 0 12	0 0 0 0 0 0 13		6 7 9 10 9 7 7 7 12
мрм	# 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	3 13 13 13 13 13 13 13 13	15 12 12 15 12 12 12 12 12	3 3 13 13 13 13	15 15 15 12 3 15 3 -3	3 3 13 3 3 3 2	2 0 2 3 15 3 2 3	00033200	0 0 2 15 3 0 6	0 0 0 3 0 0 0	000000000		00000000	7 6 7 9 10 9 7 7	MPM	# 9 2 2 2 2 2 2 * 1 9 * 1 9 *	3 3 3 3 3	15 3 15 3 15 •••••	3 13 3 13 3 13	15 12 12 15	3 13 13 3 	3 15 3 2 	0 3 0 3 3	0 3 0 13	0 0 0 12 0	0 0 0 13	0 0 0 3	000000000000000000000000000000000000000	7 9 7 12 7
••• MPM	3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	13 15 15 12 15 15 12 15 15 15 15 12	12 3 13 3 13 3 13 3 13 3 3 3 3 3 3 3 3 3	13 2 3 3 2 3 3 3 3 3 3 3 2 2 3 3 3 3 2	3 13 3 2 3 13 13 2 2 3	2 3 12 15 3 13 2 12 3 3 3 13	3 15 0 15 12 3 13 3 15 12	15 0 0 0 0 3 3 13 0 13	3 0 0 0 0 0 15 3 12 0 3	15 0 0 0 0 0 0 15 0 0 2	3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		12 7 6 7 7 9 10 9 7 12	•	2 2 2. 2	3 3 3 3	13 3 13 13	12 15 12 12	3 3 13 13	15 12 3 3	3 13 3 2	03073	0 3 0 15	0.003	0 0 15	0 0 3	000000000000000000000000000000000000000	7 9 7 12
мрм	# 6 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	2 15 3 2 15 3 2 2 2	3 3 2 3 3 3 2 3 3 3 3 3	3 12 3 13 2 15 3 13 13	15 13 15 12 15 3 12 12 12	3 3 3 3 13 13 13 13	15 0 15 15 12 13 12 3 3	0 0 0 13 12 3 0 2	0 0 0 3 13 15 0 3	0 C 0 0 3 0 15	0 0 0 0 0 0 0 3	0 0 0 0 0 0 0 0 0 0 0 0 0 15	0 0 0 0 0 0 0 0 0 0 0	7 6 7 7 9 10 9 7 12	•		•												

replaced by elements which may not be fixed.

Comparison of the MPM's will show that mechanisms 1 and 6 are equivalent, mechanisms 2 and 7 are equivalent, and mechanisms 9 and 10 are equivalent. Thus chain 48 with special elements as shown in Figure 16, may be converted into only four unique mechanisms, mechanisms 1, 2, 5, and 9.

#### Example 3

The third example will be based on chain 206 from Appendix A. The following special elements will be added:

- 1. Link 5 replaced by a pulley
- 2. Links 4 and 6 replaced by a belt
- 3. Links 8 and 9 replaced by a spring
- 4. Prism pair placed at joint 3-10
- 5. Double joint placed at position 3(2,4)

The chain resulting from the above additions would appear as shown in Figure 17. Note that the pulley need not be circular.

The loop matrix for chain 206 without the special elements added is shown in Table XXXIV. As in the previous examples, the special elements are added to form the loop matrix shown in Table XXXV. This loop matrix is then used to generate the CPM in Table XXXVI. The MPM's appear in Table XXXVII and, after comparison, show that all five mechanisms are unique.

The previous examples demonstrate the capabilities of the path matrix approach in describing and comparing kinematic chains with all types of special elements. The following chapter will show how the CPM can be applied to the development of gear trains.



Figure 17. Chain 206 with Special Elements Added

LOOP MATRIX FOR CHAIN 206

	,											
20	202	303	510	0	0	0	0	0	0	0	4	
510	303	204	305	0	0	0	0	. 0	0	0	4	
. 30.	5 206	307	510	0	0	0	0	0	0	0	4	
307	208	209	510	0	0	0	0	0	0	0	4	

### TABLE XXXV

LOOP MATRIX FOR CHAIN 206 WITH SPECIAL ELEMENTS

201	202	1616	303	1515	510	0	0	0	0	0	6
510	1515	303	1616	1312	1211	0	0	0	0	0	6
1211	1313	307	510	0	0	0	0	0	0	0	4
307	1414	510	0	0	0	0	0	0	0	0	3

# TABLE XXXVI

CPM FOR CHAIN 206 WITH SPECIAL ELEMENTS

												•
201	202	1616	303	1515	510	0	0	0	0	0	6	
510	1515	303	1616	1312	1211	0	0	0	0	0	6	
1211	1313	307	510	0	0	0	0	0	0	0	4	
307	1414	510	0	0	0	0	0	0	0	0	3	
201	202	1616	1312	1211	510	0	0	0	0	0	6	
0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
510	1515	303	1616	1312	1211	1313	307	0	0	0	8	
0	0	0	0	0	0	0	0	0	0	0	0	
1211	1313	307	1414	510	0	0	0	0	0	0	5	
201	202	1616	1312	1211	1313	307	510	0	Ō	0	8	
- 0	0	0	0	0	0	0	0	0	Ő	Õ	Ō	
. Õ	Ő	Õ	Õ	Ő	Ő	Ő	Õ	Õ	Õ	õ	Ō	
510	1515	303	1616	1212	1211	1212	307	1414	Ő	Õ	ğ	
201	202	1414	1212	1011	1212	207	1/1/	510	0	0	ó	
201	202	1010	1912	1411	1919	207	1414	510	0	0	9	

## TABLE XXXVII

# MPM'S FOR CHAIN 206 WITH SPECIAL ELEMENTS

МРм	# 2 2 2 2	1 2 2 2 2	16 16 16 16	3 13 13 13	15 12 12 12	5 5 13 13	0 0 3 3	0 0 5 14	0 0 C 5	0 0 0 0	0 0 0 0	6 6 8 9
MPM	# 2 2 2 2	2 16 16 16 16	3 13 13 13	15 12 12 12	5 5 13 13	2 2 3 3	0 0 5 14	0 0 2 5	0 0 0 2	0 0 0 0	0 0 0 0	6 6 8 9
мрм	# 3 3 3 3	3 15 16 16 16	5 13 13 13	2 12 12 12	2 5 13 13	16 15 3 3	0 0 5 14	0 0 15 5	0 C C 15	0 0 0 0	0 0 0 0	6 6 8 9
МРМ	# 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	7 5 14 5 14 5 14 14	12 5 15 5 2 5 5	13 0 3 12 2 15 2	0 0 16 13 16 3 2	0 0 13 0 13 16 16	0 0 12 0 12 13 13	0 0 13 0 13 12 12	0 0 0 0 13 13	0 0 0 0 0 0	0 0 0 0 0 0	4 3 8 5 8 9 9
мрм	#1 555555555555555555555555555555555555	C 2 15 12 3 2 15 12 2 15 2 15 2	2 3 13 14 2 3 13 2 3 2	16 16 3 0 16 16 16 16 16	3 13 0 13 13 13 14 13 13 13	15 12 0 12 12 12 0 12 12 12 12	0 0 0 0 13 0 13 13 13	0 0 0 0 3 0 3 3 3 3	C 0 0 C C 0 0 0 0 14 14		0 0 0 0 0 0 0 0 0 0 0	6 6 4 3 6 8 5 8 9 9

#### CHAPTER XI

#### STRUCTURAL SYNTHESIS OF GEAR TRAINS

Johnson and Towfigh (10) have shown that certain kinematic chains may be converted to gear trains, both compound and epicyclic. In order to be convertible to a gear train, the chain must meet the following criteria:

1. Every link in the associated linkage must belong to at least one four-link path.

2. The chain must contain at least one multiple joint (double joint, triple joint, etc).

3. Any binary link incident to a multiple joint must also be connected to another binary link by a revolute pair. Once the above criteria have been satisfied, the chain may be converted to an equivalent gear linkage. Gear trains may be formed from the gear linkage by fixing various elements with respect to ground, just as mechanisms are formed from kinematic chains by fixing various links.

To construct the equivalent gear linkage, each joint in the kinematic chain must be designated as either a rotation point (axis for a gear) or a base point (point on the edge of a gear). Certain joints are required to be either a base point or a rotation point according to the rules specified in Table XXXVIII. The remaining joints may be chosen as either a base point or a rotation point. However, as each one is chosen, the rules in Table XXXVIII must be applied to determine

### TABLE XXXVIII

## MANDATORY BASE POINTS AND ROTATION POINTS

I.	Mandatory Rotation Points										
	a. Joints between higher links b. Multiple joints										
II.	Mandatory Base Points										
	a. Joints between two binary links										
	b. Joint at other end of binary link with one rotation point										
	c. Both joints of a binary link connected to the link in b										
	d. Both joints on a binary link which connects two higher links										

## TABLE XXXIX

RULES APPLYING TO MANDATORY POINTS IN FIGURE 18

JOINT	TYPE POINT	RULE	
1-6	R	Ia	
3-6	R	Ia	
3(4,9)	R	Ib	
7-8	В	IIa	
9-10	В	IIa	
6-10	В	IIc	
5-6	В	IIc	
1-2	В	IId	
2-3	В	IId	

if other joints are affected by the choice. Where several choices exist, several gear linkages may be formed.

To illustrate this method, chain number 223 from Appendix A will be converted to a gear linkage, first, by observation of the chain itself, and second, by use of the CPM.

Figure 18 shows chain 223 with a double joint at position 3(4,9). The chain meets all criteria for conversion to a gear linkage. The mandatory rotation points and base points are shown in the figure. The rule pertaining to each mandatory point is shown in Table XXXIX. The only joints which remain free to choice are joints 6-7 and 1-8. If one of these is chosen to be a rotation point, the other automatically becomes a base point because of rule IIb. No symmetry exists for the chain as shown so that two unique gear linkages may be formed. The equivalent gear linkage formed by placing a rotation point at joint 6-7 is shown in Figure 19.

Gear trains may be formed by fixing any link containing one or more rotation points. Thus, gear trains could be formed by fixing any links of Figure 19 except links 2, 5, 8, and 10. Fixing links 4 or 9 would result in identical gear trains due to symmetry of the chain. Therefore, only five unique gear trains exist.

To apply the CPM to the development of gear trains, the following procedure would be followed:

1. Develop chains with one double joint.

2. Determine if each link appears in a path containing four links. This is done by simply taking the links in numerical order and determining if each one appears in at least one row of the CPM containing four links.



Figure 18. Chain 223 with Double Joint and Base and Rotation Points



Figure 19. Equivalent Gear Linkage for Figure 18

3. Check each row in which a double joint appears to insure that if a binary link is next to the double joint, a second binary link is next to the first.

4. Identify the mandatory rotation points

a. Any joint where type numbers of the connected links are greater than 2.

b. The double joint

5. Identify the mandatory base points.

a. Any joint connecting binary links

b. Second joint on any binary link with a rotation point at the first joint.

c. Both joints of a binary link connected to the link in b.

d. Both joints of a binary link appearing between two higher links.

6. One at a time, make each remaining joint a rotation point and apply the rules for mandatory base points. The number of possible combinations resulting will yield the number of possible gear linkages. To determine whether or not two chains are equivalent, all that is required is to show that the optional joints converted to rotation points are equivalent. If, for example, joints A and B were found to be equivalent in the original chain (by methods used in Chapter VIII), then, placing a rotation point at joint A would result in the same gear linkage as that produced by placing a rotation point at joint B.

7. Gear trains may be formed by considering each link which contains a rotation point as being fixed. Equivalent trains may be found by determining which links are equivalent (by methods used in Chapter X).

Although a computer program has not yet been written to convert kinematic chains to gear trains, it appears that the CPM contains all the information necessary to perform the steps described above.

#### CHAPTER XII

#### SUMMARY AND CONCLUSIONS

The objective of this thesis has been to present the path matrix approach to the description and analysis of kinematic chains and mechanisms. It has been shown that a kinematic chain may be completely defined by the loop matrix which, in turn, is used to generate the Chain Path Matrix. Since the Chain Path Matrix numerically describes all paths or circuits in a kinematic chain, it serves as a powerful tool in evaluating, comparing, and modifying the chain with the aid of the high speed computer. The Mechanism Path Matrix, developed from the Chain Path Matrix, uniquely describes mechanisms derived from a kinematic chain and serves as a means of comparing chains and mechanisms, again using the computer.

The computer programs developed for use in analyzing the ten-link chains were initially tested on the six and eight-link chains. The results from these tests agreed with those obtained by Hain (2,3,4,5) through many months of work. These initial tests vividly illustrated the benefits to be gained by a computerized approach to the analysis of kinematic chains.

Once the programs had been proved on the six and eight-link chains, they were applied to the ten-link chains and catalogs of the following material were obtained (9):

1. Ten-link mechanisms

- 2. Ten-link chains with 1, 2, 3, and 4 springs
- 3. Ten-link chains with 1, 2, 3, and 4 pulleys
- 4. Ten-link chains with 1, 2, 3, 4, 5, and 6 cam pairs
- 5. Ten-link chains with 1 and 2 prism pairs

6. Ten-link chains with 1 and 2 double joints.

This material provides a great deal of data for further analysis by kinematicians. Moreover, it is felt that the possibilities of the path matrix approach have only been touched upon and that further development of the concept will prove to be of great value in the type synthesis and analysis of kinematic chains.

In addition to providing material for further research, the data obtained from the ten-link chains describes a tremendous number of kinematic chains, each of which can furnish a number of mechanisms. For the designer, this data provides many approaches to a given problem and may also serve to stimulate development along lines not previously considered.

To further develop the path matrix approach, it is recommended that research be conducted in the following areas:

1. Determination of the rules necessary for the development of loop matrices for chains with crossed links. This area will become of more importance in the analysis of chains with more than ten links.

2. Use of the path concept in the structural synthesis of plane kinematic chains. This area appears promising since higher order chains may be developed from chains of a given order by adding additional loops. If this can, in fact, be done, it should eliminate the problems encountered with the crossed linkages since the CPM could be developed directly without having to form the loop matrix. 3. Use of the path concept in the structural analysis and synthesis of spatial kinematic chains. If appropriate requirements for mobility are defined for paths containing various types of joints, it appears that the synthesis of multi-loop space mechanisms should be possible.

4. Development of a means of automatically sketching chains and mechanisms represented by a CPM. This would provide a much better reference for designers than the catalogs presently developed for the ten-link chains.

5. Research into the most advantageous means of cataloging the material obtained from analysis of higher order kinematic chains so as to be of greatest benefit to the designer, research kinematician, etc. As the data on the ten-link chains now stands, the chains may be classified according to the type of elements inserted in the basic chain. However, if chains with combinations of elements are developed, the classification of the chains will require some method which facilitates retrieval of the desired information.

6. Development of a computer program to synthesize gear trains based on kinematic chains. The procedure for this has been outlined in Chapter XI. The application of such a program to the ten-link chains would provide a valuable addition to the data already obtained.

#### A SELECTED BIBLIOGRAPHY

- (1) Davies, Trevor H., and Frank Erskine Crossley. "Structural Analysis of Plane Linkages by Franke's Condensed Notation." <u>Journal of Mechanisms</u>, Vol. 1 (1966), 171-183.
- (2) Hain, Kurt. Applied Kinematics. New York: McGraw-Hill, 1967.
- (3) ....Systematik sechsgliedriger kinematischer Ketten." <u>Maschinenmarkt</u>. Würzburg, Vol. 74 (1968), No. 38, 717-723.
- (4) \_\_\_\_\_. "Die zwangläufigen achtgliedrigen Getriege mit Einfachund Mehrfachgelenken." <u>Maschinenmarkt</u>, Würzburg, Vol. 70 (1964), No. 64, 10-18.
- (5) . "Höhre Winkelubertragungen in sechs- und achtgliedrigen, zwangläufigen Getrieben." Maschinenmarkt, Würzburg, Vol. 71 (1965), No. 3, 18-23.
- (6) Woo, L. S. "Type Synthesis of Plane Linkages." <u>Transactions of the ASME</u>, <u>Journal of Engineering for Industry</u>, Vol. 89 (February, 1967), 159-172.
- (7) Dobrjanskyj, L., and F. Freudenstein. "Some Applications of Graph Theory to the Structural Analysis of Mechanisms." <u>Trans-</u> <u>actions of the ASME, Journal of Engineering for Industry</u>, Vol. 89 (February, 1967), 153-158.
- Buchsbaum, F. "Structural Classification and Type Synthesis of Mechanisms with Multiple Elements." Eng. Sc. D. dissertation, Columbia University, 1967; Publication #65-15,479, University Microfilms, Inc., Ann Arbor, Michigan, 1968, 219 pp.
- (9) Quist, F. F. <u>A Catalog of Ten-Link Mechanisms and Ten-Link Chains</u> with Springs, Pulleys, Cams, Prism Pairs, and <u>Double Joints</u>. Unpublished, Oklahoma State University, 1970.
- (10) Johnson, R. C., and K. Towfigh. "Creative Design of Epicyclic Gear Trains Using Number Synthesis of <u>Transactions</u> of <u>the</u> ASME, Paper No. 66-MD-A.
- (11) Soni, A. H., and Lee Harrisberger. <u>Applied Mechanisms</u>. Preliminary Edition, Oklahoma State University, 1969.

## APPENDICES

## APPENDIX A

## LINE DRAWINGS OF THE 230 TEN-LINK CHAINS

The line drawings shown in Figure 20 were taken from the Appendix of reference 6. The links of each chain have been numbered and minor corrections have been made in some of the drawings.



•

Figure 20. Line Drawings of Ten-Link Chains



Figure 20 (Continued)

.



Figure 20 (Continued)



Figure 20 (Continued)



Figure 20 (Continued)

•

## APPENDIX B

## LOOP MATRICES FOR TEN-LINK CHAINS WITH CROSSED LINKS

Table XL shows the loop matrices used for the ten-link chains in Group 64D of Appendix A. As mentioned in Chapter III, these loop matrices were determined by trial and error.

### TABLE XL

..

CHAIN #40	0							
	109	110	301	306	307	308		
	301	302	307	306				
	302	303	308	307				
	303	104	105	306	307	308		
		±•••	200					
CHAIN #41	1							
	310	306	105	304	303	102	101	
	310	101	102	303	307	306		
	310	309	108	307	303	102	101	
	310	101	102	303	304	309		
CUATN #1.4	2							
GRAIN #4	2 301	300	303	306	110			
	201	110	305	107	305	109	1.00	302
	301	307	305	107	305	110	102	502
	301	110	302	303	300	110		
	201	110	500	202	504			
CHAIN #43	3							
	301	306	305	304	103	102		
,	301	102	103	304	110	309	306	
	310	107	308	309	110	304	103	102
	301	102	103	304	305	308	107	
OT 1 T 1 4 1	,							
CHAIN #44	4	200	100	201	20 F			
	301 201	302	103	304 100	303	200		
	301 201	303	3UD 207	100 204	205	302		
	301 201	205	307	200	303			
	301	202	304	307	110			
CHAIN #4	5							
	301	306	307	308	109	110		
	301	110	109	308	<b>3</b> 04	105	306	
	301	302	103	304	308	109	110	
	301	110	109	308	<b>3</b> 07	302		
on								
CHAIN #40	b 201	100		00/	205			
	301	102	303	304	305			
	304	303	110	306	305			
	304	303	300	107	308			
	301	305	304	308	109			

LOOP MATRICES FOR GROUP 64D OF APPENDIX A

Note: A type number of 1 represents a binary link.

<del></del>					- <u></u>		
CHAIN	<i>#</i> 47						
		306	307	110	303	102	
	301	102	303	304	105	306	
	304	308	307	306	105		
	301	306	307	308	109		
CHAIN	#48						
	301	305	306	110	302		
	301	302	103	304	305		
	304	108	307	306	305		
	301	305	306	307	109		
CHAIN	#49						
	301	308	110	305	104	303	102
	301	102	303	104	305	106	307
	307	309	308	301			
	301	308	309	303	102		
CHAIN	<b>#</b> 50						
	301	110	309	304	303	102	
	301	102	303	304	105	306	
	306	107	308	309	110	301	
	301	110	309	308	303	102	
<del></del>							

TABLE XL (Continued)

•

•

## APPENDIX C

## SUBROUTINE LOOP

Subroutine LOOP is used to develop the CPM from the loop matrix as described in Chapter III. The variables used in subroutine LOOP are shown in Table XLI. A flow chart for the rubroutine is shown in Figure 21 and the subroutine itself appears in Table XLII.

ł

١

# TABLE XLI

.

# IDENTIFICATION OF VARIABLES USED IN SUBROUTINE LOOP

Array M	-	Chain Path Matrix (CPM)
NB	-	Number of Bars (Links) in Chain
NL	-	Number of Loops in Chain
NR	-	Number of Rows in CPM
NCOL	-	Number of Columns in CPM
NK	-	NR-NL (Number of rows formed by combining rows of Loop Matrix)
KM & LM	-	Primary row combinations used in forming rows NL+1 through NR-1 of the CPM
KS & LS	-	Secondary row combinations used in forming rows 11 through 14 (for 10-link chains) of the CPM
KT & LT	-	Row combinations required for row 15 (for 10-link chains) of the CPM

.

# TABLE XLII

## SUBROUTINE LOOP

FORTRAN	IV	GL	EVEL	1, MGD	4	LOOP	DATE = 70191	20/49/09	PAGE 0001
0001				SUBROÚ	TINE	LOOP			
0002				IMPLIC	1T 1N	TEGER+2(I-N)			
0003				COMMON	ALL	M(15,15),N8,NL,NR,NC	OL, NK, NCH, NELEN, NSTORE	· · · · · · · · · · · · · · · · · · ·	
0004		c		CUMMUN	/[00/	MALLIJJCALLIJJK31111	, , L & { 1 L J } K & ( 3 J } L   ( 3 J		
		Č	READ	IN LQ	OP\$				
0005				DO 10	[#1.N	L			
0006				A EAG (5	+1){M	{I,J},J},J=1,KCCL}			
0007			1	FORMAT	(2014	<b>)</b>			
0008		•	10	CUNIIN	U2:	N1			
0010				1500 14-		n L			
0011				00 11	1.8=1-i	NCCU			
0012				TREMEL	A.181	-GT-9000) H(IA,18)=H	(IA.[B)-9000		
0013				IPCHEL.	A, 18)	.NE.O) ISUM-ISUM+1			
0014			11	CONTIN	Ú£ 👘				
0015			12	HETAIN	cor)=	LSUM			
0016				00 200	1414	NK			
0011				13-1					
0019				Lainth	11 A.				
0020				JL#1	•	•			
0021			13	J#1					
0022				1F(1+N	L.EQ.	NR.AND. [M [K, 1].#Q.0.	OR.M(L,1).#Q.0)) GOTO 400	1	
0023				1F11+N	L.GE.	NK.AND.(M(K,1).£Q.O.	OR.M(L,1].EQ.0}) GOTO 300		
		2	STAR		1000	*			
		č			Fahr	<b>N</b> .			
		. č	ChEC	K TO S	ËE IF	ELEMENT J EXISTS IN	BOTH LOOPS		
0024			20	KN=HEK	, NCOL	3			
0025				LN=F(L	NCCL	1			
0026				IFILN.	EQ.01	GGTO 100			
0027			22	15/04	11.51	LN			
0020			21	CONTIN	1011E	erficients bere so			
0030			••	MEI+KL	']F)≐	M(K.J)			
0031			•	J=J+1					
0032				JL#JL+	1				
0033				IFINIK	eJ) • k	Q.0) GOTO 100			
0034		~		GOTO 2	2	· · ·			
		С С С	CHEC	K ÉLEM	ENTS	ON BOTH SIDES OF POS	SIBLE TRANSFER ELEMENT		
0035			30	IFCIL.	EQ.1)	GOTO 31	· ·		
0036				111.4	Q. KNI	GOTO 32			
0037				IFINIK	*1+1)	.EQ.M(L,IL-1)) GOTO	40 .		
0038				HEE+NL	*1†)=	M(K+J)			
0039				J#J+1 10/0/0	=	0 01 0010 100			
0040				11.0.11.4	9 J 3 A E 1	d'01 0010 100			
0042				doto 2	2				
0043			31	IF(J.E	0.KN)	6010 33			
0044				IFIMIK	,J+1)	.EQ.M(L,LN1) GOTO 40	)		
0045				METINL	, JL )=	M(K,J)			
0046			• •	J=3+1					
0047				IFINCK	+ J I • E	Q.0) GUTO 100			

.

# TABLE XLII (Continued)

	FORTRAN	1V G	LEVE	L 1. HOD 4	LOOP		DATE = 70191	20/49/09	PAGE 0002
	0048			JL=JL+1					•
	0049			GUTO 22				•	1
	0050		، د	COTO 100	G.M(L,IL-II) 6010 4	U	· .		
	0052		33	3 [F{M(K,1).E	G.M(L.LN)) GOTO 40				
	0053			GOTO 100	· ···			•	
			C						· · ·
			C NEX C ELI C TR/	CT ELEMENT IN Ement was com Ansfer to sec	FIRST LOOP IS COMM MON. IF IT WAS, PR OND LOOP.	ON TO CCEED	BUTH LOOPS. CHECK To Next Element in	TU SEE IF LAST FIRST LOOP. IF NO	)T
	0054		Ŭ 40	]   F  J.EQ.1}	GCTC 41				
	0055			IFIIL .EQ.LN	) GOTO 42				
	0056			IF(M(K,J-1)	-NE.M(L,IL+1)) GOTO	50			
	0057			 	0 01 6010 100	,		· .	
	0059			GUTO 22					
	0060		41	L IF(IL.EQ.LN	1 GBTQ 43		•		
	0061			IFIMIK.KNI.	NE-P(L, [L+1]) GCTO	50			
	0063			1F(M(K,J).E	Q.01 GUTO 100				
	0065		4	2 IF(M(K,J~1)	.NE.#(L.1)) GOTO 50				
	0066			J=J+1					
	0067			1F[F(K,J).E	G.O) GCTC 100				
	0069		43	3 IF(M(K,KN).	NE.M(L.1)) GOTO 50				
	007C			J≠J+1	-				
	0071			1F(M(K,J).E	Q.C) GUTO 100				
	0012		c	GUIU 22					
			C TR/	ANSFER TO SEC	OND LOUP				
	0073		5. 50	MEI+NL+JL)=	M(L,1L)				
	0074		51	L JL=JL+1					
	0076			16(0)	50.01 H.al				
	0077		52	2 DO 53 JalaK	N				
	0078			IF(MIL,IL).	EC.M(K,J)) GUTE 60				
	0079		53	CONTINUE				en e	
	0081			60TH 51	PILILI				
			С	0010 34					1
			C CHE	CK ELEMENTS	ON BOTH SIDES OF PE	SSIBLE	TRANSFER ELEMENT		
	0082		ت 61	1 (F(J.EC.1)	GOTO 61				
	0083			IF(IL.EQ.LN	J GOTO 62				
	0084			1F(M(L, 1L+1	1.E4.M(K,J-1)} GOTO	70	· · · · · ·		
	0085		6	6510-100   1611 .60.18	EA 0103 43				
	0087			1F(M(L.1L+1	).EC.M(K.KN)) ECTO	70			
	8800			GUTU 100					
	0089		62	2 IF(M(L,1).E	Q.M(K,J-1)) GOTO 7C		· · · · · · · · · · · · · · · · · · ·		
	0090		6	3 1F(M(L+1)_F	0.#(K.KN)) GUTO 70				· · · · ·
	0092			GETC 100	•••••				
	· · ·		C NE	CT ELEMENT IN	SECOND LOCO IS CON		BOTH LCOPS. TRAN	SEER BACK TO FIRST	
. 8	en in State		C NL	AT CLEMENT IN			Dorn Codesi - jak	WIER MACK TO TRUT	
	· · .								
	FURTRAN	iv G	LEVE	L 1, MEC 4	LCOP		DATE = 70191	20/49/09	PAGE 0003
	P		-				•	•	
			C LUL	. 41				· •	
	C093		~ 7(	) IF(M(K,J).E	C.M(I+NL,1)) GUTO 1	10			
	C094			MII+NL,JLI=	H(K,J)				
	0095			.J#J+[  C/N/#.J\ C	C 01 1-1				
	0057			JL=JL+1	C.07 3-1				
	0098			GOTC 70					
	0055		300	) K=KS([)					
	0100			60TO 20					
· .	0102		400	K=KI(IS)					
	0103			L=LT(IS)					
	0104			15#15+1 16/18 CT CA	COTO 100				
	0106			GCT0.13	3510 100				
	0107		100	JL=1					
	0108		110	DO 111 J=JL	NCOL				
	0104		111	<pre>Mil+NL_JJ=0 Mil+NL_NCOI</pre>					
	0111		200	CONTINUE	· · · · ·				
	6112			RETURN					



Note: KN is number

2

?

K J+1 = L II-1

KJ-1 = LTL+1

(I+NL) JL - L

JL+- JL+1

IL+ IL+ 1

κ, = (I+N)

Complete row

I+NL with O

Last element of

row I+NL equals

JL-1

 $(\mathbf{1})$ 

ю

YES

YES

NO

YES

of non-zero elements in row K

Figure 21. Flow Chart for Subroutine LOOP
#### APPENDIX D

#### SUBROUTINE MECH

Subroutine MECH develops the MPM's for mechanisms formed from a kinematic chain as described in Chapter IV. Most of the variables used in MECH are the same as those used in Subroutine LOOP. The only variables used in LOOP are MY, which holds the CPM reduced to an array of link numbers, and MX, which holds the MPM's that are generated. The flow chart for Subroutine MECH appears in Figure 22 and the subroutine itself in Table XLIII.

### TABLE XLIII

#### SUBROUTINE MECH

1		and the second second second					
FORTRAN IV	G LEVEL	1, MOD 4	MECH	DATE -	70192	09/43/16	PAGE 0001
0001		SUBROUTINE NEC	H L L				
0002		IMPLICIT INTEG	ER#2[[-N]				
0003		COMMON/ALL/MIT	5.20) . NB.NL . NR . NC	OL . NK . NCH . NEL	LEM		
0005		CONMONTICHT	10.15.151				
0004		DIMENSION MYIL	5.151				
0003	r	DIALASTON ATTE					
	č sto	RE M IN MY AFTER	R REDUCING TO LIM	IK NUMBERS			
	C						
0006		DO 11 [=1,NR					
0007		NE #M(I + NCOL)					
0008		IF(NE.EQ.C) GC	TO 13				
0009		DO 12 J=NE+NCO	L				
0010	12	WA(I*1)=W(I*1)					
0011		DO 10 J=1,NE					
0012	10	HY(1,J)=H(1,J)	-(M(I+J)/100)+100	1			
0013		GOTO 11					
0014	13	MY(I.1)=0					
0015		HY(1.2)=0					
0016	11	CONTINUE					
0010	· · ·	00.111100					
	C DEV	ELOP MECHANISM	MATRICES				
	Ĺ						
0017		DO 100 I=1,NB					
	C SEA	RCH FOR LINK 1	IN EACH ROW OF MY	,			
	L.	1 - 1					
0018							
0019		3-1					
0020	2°#	K#1					
0021		NJ=L					
0022	22	IFEMY(J+KI+EU+	11 6010 30				
0023		K=K+1					
0024		IF(MY(J+K)+NE+	D) GOTO 22				
0025		j=j+1					
0026		IF(J.GT.NR) GD	10 100				
0027		GOTO 21					
	Ç						
	C STO	RE ROW IN MX ST	ARTING WITH LINK	1			
	С.,					•	
0029	30	MX(1,L,K1)=M(J	(K)				
0029		K=K+1					
00 30		K1=K1+1					
3031		IF(M(J.K).EQ.0	) K=1				
0032		IF(M(J.K).EQ.M	X(I,L,1)) GOTO 40	)			
0033		GOTO 30					
	c						_
	C ALL C M.	LINKS HAVE BEE	N STORED IN MX.	COMPLETE ROW	WITH CORRE	ESPONDING PORTION UP	•
0004	۰, ۲	00 41 Hart M	DI .				
0034	40	00 91 JI-NI(NU)					
0035	41	MALLILJJLJ#MLJ					
0036		L=L+1					
0037		J=J+1					
0038		IFIJ.GT.NKJ GU	10 100				
0039	-	GOTO 21					
r	C.				TODEN 14		
	C ALL C	RUWS CUNTAININ	P FINK I HAAG DED	IN PROPERLY S	1050 18 67		

FORTRAN IV G LEVEL 1. MOD 4

.

MECH

DATE = 70192

09/43/16

PAGE 0002

0040 100 MX(I+1,NCOL-1)=L 0041 NCH=NB 0042 RETURN 0063 END



Figure 22. Flow Chart for Subroutine MECH

#### APPENDIX E

### SUBROUTINE MECCOM

Subroutine MECCOM compares MPM's as described in Chapter IV. A flow chart for this subroutine is shown in Figure 23. A list of the variables used in MECCOM appears in Table XLIV and the subroutine itself is in Table XLV.

### TABLE XLIV

## VARIABLES USED IN SUBROUTINE MECCOM

MX	-	Array of MPM's
NCH	-	Number of MPM's to be compared
IEQUIV	-	Array indicating equivalent MPM's
ISAMEM	-	Array which indicates whether an MPM is already equivalent to another MPM
ISAMEY	-	Array which indicates whether a row is already equivalent to another row
IHOLD	-	Array used in cycling rows
ITAG	-	Counter used to determine how many times a row has been cycled
JSTORE	-	Array in which descriptions of unique chains are stores
NSTORE	-	Number of unique chains stored in JSTORE
All variable conjunction	es i wit	n COMMON/ALL/ are described in Appendix C in h Subroutine LOOP.

•

102

## TABLE XLV

# SUBROUTINE MECCOM

FORTR	VN TA	GL	EVEL	1. HO	D 4	MECCOM	DATE = 70191	20/49/09	PAGE ODQĮ
0001				SUBRO	UTINE ME	CCOM			1
0002		Λ.		INPLI	CIT INTO	EGER#2(1-N)			
0003				CONNO	N/ALL/M	LIS. 151. NR. NL. NR. NCM	NK NCH. NELEN. NSTOR	F	
0004				COMMO	N/MEC/IS	EDHIV/50-50)	, including the conjustion		
0005				CONNO	N/DOI NY	M7/18.183.M71/18.18	M7 14 18. 181	161	
0002				6 M7 64	IE IEL A	NATISTICS 181	1455 138 131 4453( 13)	121 101 412 45 7 300 304	
0004				COMMO	43473141 43473141	11110051100 41 ITCHD	100 41		
0000				DINCH	R/UDALL/	VUSIONELLUVIUJUJUEPP	100101		
0007		~		DIMEN	210M 121	ATER 1201   13AT ET 1201   1	HULDE201		
		L	CUN	VERI I	U ARKATS	S OF LINK TYPES			
0008				LTAG	0 .				
0009				00 10	1#1+NCP	H			
0010			-	DO 5	J= L + NCH	_			
0011			5	LEGUI	A(1'1)±0	•			
0012				I SAME	NEI)=C		1		
0013				NROW=	HX(1,1,1	NCOL-11-1			
0014				MXCI,	t-NCCL-1	LINACH			
0015				00 10	J=1,NRC	DW .			
0016				NCLN	MXE1,J'H	NCOLI			
. 0017				DO 10	K=[ NCI	LM			
0018			10	MX{I,	J,K}=HX{	[1,J,K}/100			
0019				00 11	I=1;NR				
0020			11	I SARE	V(I)+0				
		c	CCH	PARE E	ÀCH HECH	HANISH WITH ALL OTHER	HECHANISHS		
0021				[X=1					
0022			1.9	IYelx	÷1 '				
<i>F</i>		C	CHE	CKTÓ	SEG LE H	AXLIVE IS ALREADY FOR	I) VALENT		
0023			20	IFILS	AMENILY	.EQ.11 OCTO 70			
		C	CHE	CK TO	SEE IF F	HALLY) HAS SAME NUMB	R OF ROWS AS MALTA	1	
0024			••••	TECHX	112.1.0	01-11-NE-MXI [Y-1-NC	L-111 GOTO 70		
		c	COM	PARE F	ACH ROW	OF MALIAN WITH BOWS	GE MX(IV)		
6025		. •		TYROW	=1 .				
0026				I VOAL	-1				
0027			10	IEIMY		NUMBER OF AND AND AND AND			
0021		c	C 11 E		CEE IE	WYITY TYDOWS TO ALDE	DY FOUTVALENT TO A	ADM IN MY/LY)	
0078			UNE	16/16	JEC IF T	POUL BO IL COTO DO	UT EQUITALENT TO A		
0020		· ~	C.C.H	15133	A FARNTS	DE MYTER ENDOWS WET		NUMBER OF STON	
0036			CUR	TARE C	TENEUI3	OF HALLAJIABURI HIII	I HALLIJI KUNJ IN PO	AWARD DIRECTION	
0027				TACUL	-1 //u //un/	NU THEOL & NO MULTIN P	MON EXCOLLS COTO SI		
0030			40	LE LEAA	LIAJIAK	CHAINCHEISUCSWY(IIII	KUNITACULII GUID I		
0031				IXCUL	= 1X UUL + 1	L WW/IN INDON NECLAR C	70 40		
0032				10 11 1	GUL.LÉ.I	MALIA,IXKUWANLULJA G	10 40	•	
0033				12445	TELTKOW:	1=1			
PE UU		-		0616	100				
		C	INC	KRAENT	1.4				
0035			/0	14= 14	*1				
0036				LEEIY	-GT - NCH	F GUTC 80			
0037		_		GOTO	20				
		· C	INC	REMENT	1X				
0038			80	EX#EX	+1				
0039				TELIX	.GT.NCH	-1) GOTO 21C			
0040				GOTO	19	· .			
		C	INC	REMENT	I YROW				
0041			90	IVRON	= IYROW+1	1			
0042				I TAG=	0.				
0043				TECTA	RON.GT.I	MX(EX,1,NCOL-1)) GUT	100		
0044				GOTO	30				
		C	INC	REMENT	I XROW				
0045		-	100	IXROW	=IXROH+)	1			
0046				IFUX	ROW.GT.I	MX(IX,I,NCOL-1)) GOT	200		

.

0047							
		IYROW=1					
0048		GOTO 30					
	C COM	PARE MX(IY, IYRO	W) IN REVERSE				
0049	- 110	IXCOL=1					
0050		IYCOL=MX(IY+IY	ROH,NCOL}+1				
0051		MX(IY, IYROW, IY	COLJ=MX(IY,[YROW;L	3			
0052	111	IFEMX(1X,1XROW	IXCOL).NE.MX(IY)I	VROW, IVCOLIJ GOTO 120			
0053		IXCCL=IXCCL+1					
0054		IYCOL=IYCCL-1					
0055	,	IF(IXCOL.LE.MX	(IX,IXRO%,NCCL)) G	OTO 111			
0056	()	ISAMEY(IYROW)=	L				
0057		GOTO 100					
0058	120	NE=MX(IY,IYROW	,NCOL )				
0059		IF ( ITAG. EQ.NEL	EH-L) GOTO 90				
0060		00 130 IA=2,NE					
0061		IF(MX(IY, IYROW	, [A].Eú.MY(IY, IYRO	Wy1)) GOTO 131			
0C62	130	CONTINUE		•			
0063		GOTO 90					
	C CYC	LE ROW TO NEXT	ELEMENI				
0064	131	ISTOP=IA				•.	
0065		1=HL					
0066	132	IHOLD(JH)=MX()	Y, IYROW, IA)				
0067		IA=IA+1	•				
0068		IF(IA.EQ.ISTOP	1 GOTQ. 133				
0069		IF(IA.GT.NE) L	A=1				
0070		1+H6=HF		•			
0071		GGTO 132					
0072	133	00 134 IH=1,JH					
0073	134	MXE IY, IYROW, IH	)=1HGLO(IH)			/	
0074		1TAG=LTAG+1	· •		•		
0075		IXCOL=1		•			
0076		GUTO 40					
	C ALL	ROWS HAVE BEEN	CCMPARED. CHECK	TO SEE IF MATRICES ARE	EQUIVALENT.	•	
0077	200	NROW-MX(IX,L,N	COL-11				· ·
0078		ISUM=0					
0079		00 201 I=1,NRO	W. C.				
0080		IFIISAMEVII).E	Q.L) ISUM=ISUM+1			•	
0081	201	ISAMEY()=0					1 A.
0082		IFIISUH EQ .NRO	W) GOTO 220				
6083		GOTO 70		•			
0084	220	IEQUIV(IX, IV)=	IY				
0085		ISAMÉH(1Y)=1					
0086		DO 1000 [#=1,N	ELEM				
0087	1000	JSTORELLY, IWI=	0				
0088		GCTC 70					
	C hRI	TE EQUIVALENCE					
0089	210	CALL PRINT					
0090		DO 1001 IW=2,N	Сн				
C091		IFIJSTORE(IN,1	).EC.0) GCTC 1002				
0092	1001	CONT INUE					
0093		NSTORE=NCH					
0054		RETURN					
0095	1002	IZERO=IW					
0096		1+P1=1++1 .					
0097		DO 1006 IN=15P	1,NCH				
0098		IFIJSTORE (IW.1	).NE.0) GOTO 1004				
0099		GOTO 1006					
0100	1004	00 1005 1X+1.N	ELEM	the second se			
			1. Sec. 1. Sec				
				and the second			
				11 [8] .			
	IN G LEVEL	1. #00 4	MECCEN	DATE = 70191	20/49/09	94	IGE 0003
ON TONN							
0101	1005	JSTORE(IZERO.I	X)=JSTORE(IN-IX)				
0102		IZERC=IZERO+1		1.1			
0103	1006	CONTINUE		e tao ga			
		NSTORE=1ZERO-1					
0104							
0104		RETURN					
0104		RETURN END					
0104 0105 0106		RETURN ENO					

101 -	1005	1210KF(1%FK0*1%)*72
102		IZERC=IZERO+1
103	1006	CONTINUE
104		NSTORE=1ZERO-1
105		RETURN
106		ENO

.



Figure 23. Flow Chart for Subroutine MECCOM

### APPENDIX F

### PROGRAM FOR DETERMINING THE TEN-LINK MECHANISMS

The program shown in Table XLVI was used to generate the ten-link mechanisms. The input data required for the program (and all other programs) is described in Table XLVII.

## TABLE XLVI

# PROGRAM FOR DETERMINING TEN-LINK MECHANISMS

ORTRAN IN	G LEVEL	1, MOD	4	MAIN	DATE = 70191	20/59/33	PAGE 0001
	Ċ						
	C TEN	-LINK C	HAINS				
	C .		~~		· · ·		
	L CEF	INITION	DF INPUT	EATA			
	Ľ,	NI		E 10005			
	ř		NUMBER O	NE RADC			
	č	NG	NUMBER C	VE ROWS IN LOOP N	ATRIX		
·	č	NCOL -	NUMBER O	F COLUMNS IN LOC	P MATRIX		
	č	NK -	NUMBER O	F LOOPS IN ADDIT	ION TO BASIC LOOPS		
	C	NC -	NUNBER C	F CHAINS			
	C	KHELH-	SEQUENCE	S FOR DERIVING L	00P S		
	C	KSELS-	SECONDAR	Y SEQUENCES FOR	DERIVING LOOPS		
	C	KTELT-	TERTIARY	SEQUENCES FOR C	ERIVING LOOPS		
	C						
	C****	******	********	*************	******************	***************	•
0001		IMPLIC	IT INTEGE	R#2(1-N)			
0002		COMMON	ALL/MILS	JZUJINBINLINKINC	ULINKINCHINELEM		
0004		COMMON	/ 200/ 8711	.17 (LM(117 (N3(11)	1211111KI (377LI (37		
0004		CONNON	/	AL . T. TROW	•		
0006		COMMON	INFC /IFOL	HV(10.10)			
0007		READ(5	ALL NLINE	NR.NCOL.NC			
0008		NK=NR-	NL				
0009	1	FORMAT	(1515)				
0010		READ (5	,13CKM(13	,1=1,NK)			
0011		READ(5	,13(LM(I)	,[=1+NK}			
0012		NELEN=	1				
0013		NS=NK-	NL.				
0014		READIS	,1)(KS(1)	+I=NS+NK}			
0015		READIS	+11(LS(1)	L=NS,NKI			
0012		READIS		+1=1+2/			
0011		TTOTAL	-0 	*******			
0018		CALL D					
0020		00 2 1	1.86				
0021		CALL	002				
0022		CALL M	ECH				
0023		CALL M	ECCOM	,			
0024		IFILRO	4.GT.25)	CALL PRINT1			
0025	2	CONTIN	UE				
0026		WRITE	6,4) ITOT	AL			
0027	4	FORMAT	LI OTOTAL	NUMBER OF 10-LIN	K MECHANISMS - +,		
		1 15)					
0028		WRITE	6,10001				
0029	1000	FORMAT	(*1*)				
0030		STOP					
0031		END					

0003		1, 800 4	PRINT	DATE = 70191	20/59/33	PAGE 000
9001		SUBROUTINE PRIM	NT			
0002		IMPLICIT INTEG	ER+2(1-N)			
0003		COMMON/ALL/HIL!	5,20], N8, NL, NR, NC	DL, NK, NCH, NELEN		
0.004		COMMON/MEC/IEQ	U[V(10,10)	•		
0005		COMMON/PRI/ITO	TAL,1,IROW			
0006		DIMENSION ANUM	(10),EQUIV(10)			
0007		DATA BLANK / 1	/			
0008		OATA EX/ * X*/				
0009		DATA ANUM/* 1*	4 21.1 31.1 41.1	51.1 61.1 71.1 81.1 91.	*10*/	
0010	1	FORMAT( + + - T11	10(A2.2X1)			
0011	2	FORMATETIO.11	•1 • • • • • •			
0012	3	FORMATILY.68( -	-1))			
0013	Ă	FORMATCH+ 1.1	3. 1 1 1			
0014	;	EORMAT/ ++ . TO	11.10/11 11.15			
0015		509MAT( 4.1. 151.	.114.759.121			
0016	•	15UM-10				•
0017		UBTTE(4 31				
0010		UNITE (0,3/				
0010		WRITELOFT				
0019		WALLELO,21				
0020		WRIIE(0,6) 1				
0021		00 20 EX=1,10				
0022		00 10 1V=1,10				
0023		EQUIVELYJABLAN	Κ.	· · · · ·		
0024		IFCIEQUIVCIX, IN	Y).E0.99) EQUIV(1	K)=EX		
0025		TELIEONTALIX'I.	Y).EQ.1Y)EQUIV(1Y	}=ANUH(IX)		
0026	~ -	IF(EQUIV(IY).N	E.BLANK) ISUM=ISU	H-1		
0027	10	CONTINUE				
0028		WRITE(6,1) (EQ)	UIV(J),J=1,10)			
0029	20	CONTINUE				
0030		WRITE(4.8) ISU	No. 1			
		ITOTAL - I TOTAL AL	E @ 1 100			
0031		TIOINC-TIOINCA:	1904			
0031 0032		IRGN=IRON+1	1204			
0031 0032 0033		IROW=IROW+1 AETURN	1304			
0031 0032 0033		IRGN=IRGN+1 AETURN END	1300			
0031 0032 0033 0033		IRCH-IRCH+1 AETURN END				
0031 0032 0033 0033		IRCH-IRCH+1 AETURN END		с. с. 	<b>.</b> .	
0031 0032 0033 0034 FORTRAN IV	G LEVEL	IROW-IROW+1 AETURN END 1, MOD 4	PRENTL	DATE - 70191	20/59/33	PAGE 000
0031 0032 0033 0034 Fortran IV 0001	¢ LEVEL	IRON-IRON+1 AETURN END 1, MGD 4 SUBROUTINE PR11	PRENTI	DATE - 70191	20/59/33	PAGE 000
0031 0032 0033 0033 0034 FORTRAN 1V 0001	G LEVEL	IRQUEIROUNI AETURN END 1, MGD 4 SUBROUTINE PRII	PR[NT] PR[NT] ER2(1-N)	DATE - 70191	20/59/33	PAGE 000
0031 0032 0033 0034 FORTRAN IV 0001 0002 0003	G LEVEL	IRCU-IRCUNI AETURN END 1, NGD 4 SUBROUTINE PRIJ IMPLICIT INTEGI COMMON/PRI/ICO	PRENT1 PRENT1 NT1 ER02(1-N) TAL:E:ER04	DATE - 70191	20/59/33	PAGE 000
0031 0032 0033 0034 FORTRAN IV 0001 0002 0003 0004	& LEVEL	IT ALL INCOMENTATION OF A CONTRACT OF A CONT	PRENT1 NT1 ER42(1-N) IAL, E, EROW 100)	DATE - 70191	20/59/33	PAGE COO
0031 0032 0033 0034 FORTRAN 1V 0001 0002 0003 0004 0005	G LEVEL	IL ALL INCOMENTATION AND ALL ALL ALL ALL ALL ALL ALL ALL ALL AL	PRINT1 PRINT1 ER2(1-N) TAL,I,IROW (10) * 2*.* 3*.* 4*.4	DATE = 70191 5*.* 4*.* 7*.* 8*.* 9*.	20/59/33	PAGE 000
0031 0032 0033 0034 FORTRAN IV 0001 0002 0003 0004 0005 0006	G LEVEL	II ON THE FIGHT I RETURN END 1, NOD 4 SUBROUTINE PRIJ IMPLICIT INTEG COMMON/PRIJIO DIMENSION ANUM/ PATA ANUM/ 11 FORMAT(++,+1)	PRINT1 PRINT1 ER92(1-N) TAL,I,IROW (10) , 21, 31, 4., 4	DATE = 70101 5*,* 6*,* 7*,* 8*,* 9*,	20/59/33	PAGE 000
0031 0032 0033 0034 FORTRAN 1V 0001 0002 0003 0004 0005 0006 0006	G LEVEL	I GUNIE I GUNI AETURN END I, MOD 4 SUBROUTINE PRII IMPLICIT INTEG COMMON/PRI/ITO DIMENSION ANUM/ DATA ANUM/* 1* FORMAT(***,T11) FORMAT(***,T12)	PRENT1 NT1 EA2(1-N) TAL.E.(ROW (10) .00(A2,2X)) 4 1)	DATE - 70191 5*,* 6*,* 7*,* 8*,* 9*,	20/59/33 • 10•/	PAGE 000
0031 0032 0033 0034 FORTRAN IV 0001 0002 0003 0004 0005 0006 0007 0008	G LEVEL	II ADD INC. INC. I AETURN END I, MOD 4 SUBROUTINE PRIJ IMPLICIT INTEG COMMON/PRIJIO DIMENSION ANUM/ PATA ANUM/* 1: FORMAT(I).110 FORMAT(I).110	PRINT1 PRINT1 ER92(1-N) TAL.I.IROW (10) , 2:, 3:, 4:, 10(A2,2X) 1 :) 10(A2,2X) 1 :) 10(A2,2X)	DATE - 70191 5*,* 6*,* 7*,* 8*,* 9*,	20/5 <b>9/33</b> • 10•/	PAGE 000
0031 0032 0033 0034 FORTRAN IV 0001 0002 0003 0004 0005 0006 0007 0008 0009	G LEVEL	IRCUM-IRCUMLI RETURN END 1, MOD 4 SUBROUTINE PRII IMPLICIT INTEG COMMON/PRI/ITO DIMENSION ANUM/ PORMAT(*1*,711 FORMAT(*1*,712 FORMAT(*1*,712) FORMAT(*1*,712) FORMAT(*1*,712)	PRINT1 PRINT1 ER42(1-N) IAL,I,IROW (10) ,* 2*,* 3*,* 4*,* ,10(A2,2X)) * 1) //,T25,*(LINK NUMB) !*,151,*1 (NICOM)	DATE = 70191 5*,* 6*,* 7*,* 8*,* 9*, 5*,* 6*,* 7*,* 8*,* 9*,	20/59/33 • 10•/	PAGE COO
0031 0032 0033 0034 FORTRAN 1V 0001 0002 0003 0004 0005 0006 0007 0008 0009 0010	G LEVEL 1 2 4 5	II AGUATIROWAL AETURN END I, MOD 4 SUBROUTINE PRII IMPLICIT INTEG COMMON/PRI/IO DIMENSION ANUM/ ATA ANUM/* 1', FORMAT(**,TIL FORMAT(**,TIL) FORMAT(**,CMAIN WRITE(6.4)	PRINT1 PRINT1 ER2(1-N) TAL,I,IROW (10) , 2:, 3:, 4:, , 10(A2,2X)) ; 1) ; 1)	DATE = 70191 5*,* 6*,* 7*,* 8*,* 9*, ER*3 E MECHANISMS*3	20/59/33	PAGE 000
0031 0032 0033 0034 FORTRAN IV 0001 0002 0003 0004 0005 0006 0005 0008 0009 0010	G LEVEL 1 2 4 5	IT ONLE I TOUR I RECHTING HI RETURN END I, MOD 4 SUBROUTINE PRIJ IMPLICIT INTEG COMMON/PRIJIO DIMENSION ANUM/ PORMAT(10,110) FORMAT(10,110) FORMAT(10,110) FORMAT(10,110) FORMAT(10,10) FORMAT(10,10) MRITE(6,4)	PRINT1 PRINT1 ER92(1-N) TAL.I.IROW (10) • 2'. 3'. 4'. 10(A2,2X)) • 1) //,T25. LINK NUMB   •,T51.   UNIQU	DATE = 70101 5°,° 6°,° 7',° 8°,° 9', ER°) E MECHANISMS°)	20/59/33 • 10•/	PAGE 000
0031 0032 0033 0034 FORTRAN 1V 0001 0002 0003 0004 0005 0006 0006 0009 0009 0012	G LEVEL	I, MOD 4 SUBROUTINE PRII IMPLICIT INTEG COMMON/PRI/ITO DIMENSION ANUM/* 1* FORMAT(***,TIL FORMAT(***,TIL FORMAT(***(ALAN) WRITE(6,2) WRITE(6,2)	PRINT1 NT1 ER#2(1-N) IAL.I.IROW (10) * 2'.* 3'.* 4'.* .101A2.2XJ) * 1) //.T25.*LINK NUMB [*.T51.*] UNIQU	DATE = 70191 5°,° 6°,° 7°,° 8°,° 9°, ER°J E MECHANESMS°J	20/59/33 • 10•/	PAGE 000
0031 0032 0033 0034 FORTRAN 1V 0001 0002 0003 0004 0005 0006 0007 0008 0009 0012 0012 0012	G LEVEL 1 2 4 5	II GIAL I GUMAL TI REGUITADINI RETURN END I, MOD 4 SUBROUTINE PRIJ IMPLICIT INTEG COMMON/PRIJIO DIMENSION ANUM/ DATA ANUM/ FORMATIIO.II( FORMATIIO.II( FORMATIIO.II( FORMATIIO.COM WRITE(6,2) WRITE(6,2) WRITE(6,2) WRITE(6,2)	PRINT1 PRINT1 ER2(1-N) TAL.I.IRON (10) , 2., 3., 4., 10(A2,2X) ; 1) , 10(A2,2X) ; 1) , 10(A,1,10) H(IA),IA=1,10)	DATE - 70191 5°,° 6°,° 7',° 8°,° 9°, ER') E MECHANISHS')	20/59/33 • 10•/	PAGE 000
0031 0032 0033 0034 FORTRAN IV 0001 0004 0005 0004 0005 0006 0009 0009 0012 0012 0013	G LEVEL	IT CHARTER IN THE STORE I TO THE STORE I TO THE STORE I TO STORE S	PRINT1 PRINT1 ER42(1-N) IAL.I.IROW (10) * 2*,* 3*,* 4*,* * 10(A2,2X)) * 1) //,T25,*LINK NUMB: (*,T51,*[ UNIQUE N(IA),IA=1,10)	DATE = 70191 5*,* 6*,* 7*,* 8*,* 9*, ER*3 E MECHANESMS*3	20/59/33 • 10•/	PAGE COO
0031 0032 0033 0034 FORTRAN 1V 0001 0002 0003 0004 0005 0006 0007 0008 0009 CC1C 0011 0012 0013 0014	G LEVEL 1 2 4 5	I. AGD 4 SUBROUTINE PRIJ IMPLICIT INTEG COMMON/PRIJIO DIMENSION ANUM/ DATA ANUM/* 1* FORMAT(**,TIL) FORMAT(**,TIL) FORMAT(**,TIL) WRITE(6,2) WRITE(6,5) IROM=1 COTHOL	PRINTI PRINTI ER2(1-N) TAL,I,IROU (10) ,* 2',* 3*,* 4*,* 10(A2,2X)) * ; *); 10(A2,2X) * ; *); 10(A2,2X	DATE = 70191 5*,* 6*,* 7*,* 8*,* 9*, ER*) E MECHANESMS*)	20/59/33 • 10•/	PAGE 000
0031 0032 0033 0034 FORTRAN IV 0001 0002 0003 0004 0005 0006 0007 0008 0009 0010 0012 0012 0013 0014 0015	G LEVEL 1 2 4 5	IT CIAL I ROMAL IR CUMINE ROMAL RETURN END I, MOD 4 SUBROUTINE PRIJ IMPLICIT INTEG COMMON/PRIJITO DIMENSION ANUM/ PORMAT(10,110 FORMAT(10,110) FORMAT(10,110) FORMAT(10,110) WRITE(6,40) WRITE(6,110)	PRINT1 PRINT1 ER92(1-N) TAL,I,IROW (10) * 2*,* 3*,* 4*,* 10(A2,2X)) * 1) //,T25,*LINK NUMB [*,T51,*] UNIQU N(IA),IA=1,10)	DATE = 70101 5°,° 6°,° 7',° 8°,° 9', ER°} E MECHANISMS°)	20/59/33 *10*/	PAGE 000

#### TABLE XLVII

#### INPUT DATA FOR COMPUTER PROGRAMS

Card 1 NL Number of loops in chains NB Number of bars (links) in chains NR Number of rows in CPM NCOL Number of columns in CPM \_ NC Number of chains to be processed Data is read in with FORMAT (1515). Card 2 KM(1)Card 3 LM(I)KM and LM are the rows to be combined in forming other rows of the CPM. For the ten-link chains, to form row 5, rows 1 and 2 of the CPM are combined. Therefore, KM(1) would be 1 and LM(1)would be 2. Table V shows all combinations used for the ten-link chains. The data on cards 2 and 3 is read in with FORMAT (1515). Card 4 KS(1)Card 5 LS(I)KS and LS are the second row combinations to be used if the primary combinations fail. For the ten-link chains, the secondary combinations are required only for rows 9 through 15 of the CPM, therefore, only 7 entries are required on each card. The data on cards 4 and 5 is read in with FORMAT (1515). Card 6 KT(1)Card 7 LT(I)KT and LT are additional row combinations used only for row 15 of the CPM for ten-link chains. They are read in with FORMAT (15I5).Cards 8 -These cards are read in by Subroutine LOOP and furnish the loop matrices. For the ten-link chains, 4 cards are required for each chain, one card per loop. Each card provides one row of the loop matrix. (Do not insert number showing the count of the elements in the row). These cards are read in with FORMAT (2014).

The	input	card	s for	the	10-1ir	nk cł	ains	appea	ired as	fol	lows:		
Card	1)	4	10	15	15	230							
Card	2)	1	1	4	2	2	3	1	4	10	8	5	
Card	3)	2	3	1	3	4	4	8	5	1	4	10	
Card	4)	5	7	3	2	6							
Card	5)	3	2	7	10	9							
Card	6)	8	4	3	2	1							
Card	7)	7	11	12	13	14							
Card	8)	301	312	310	309								
Card	9)	302	103	104	305	310							
Card	10)	305	306	309	310								
Card	11)	306	107	108	301	309						·	
NOTE	: Th	e pro	grams	were	e set u 1 (as	ip fo	or bir Tranka	nary 1	inks h	eing	repre	sented	lЪ

OTE: The programs were set up for binary links being represented by a type number of 1 (as in Franke's notation) instead of 2. Therefore, the numbers 103 and 104 on card 9 would represent binary links 3 and 4.

•

### APPENDIX G

### PROGRAM FOR DETERMINING THE TEN-LINK CHAINS WITH SPRINGS

The complete program for determining the ten-link chains with springs is shown in Table XLVIII. Flow charts for Subroutines SPRLOC and SPRNG3 appear in Figures 24 and 25 respectively. This program serves as the model for all programs which follow.

FCRTRAN	IV G	LEVEL	. 1, MQD	-4	HAIN	BATE	= 70191	20/49/09	PAC	JE 0001
		C								
		C 1EM	HLINK C	HAINS						
		C CEF	INITION	OF INPUT	CATA					
		C C	NI -		10085					-
		č	NB -	NUMBER OF	BARS					
		ç	NA -	NUMBER OF	ROWS IN LECP	MATRIX				
		ĉ	NK -	NUMBER OF	LOLUMNS IN L	UUP MAIKIX	10085			
		č	NC	NUMBER OF	CHAINS					
		c	KNGLP-	SECUENCES	FUR DERIVING	LOOPS		/		
		C C	KSELS-	SECONDARY	SEQUENCES FO	P DERIVING LOOP	PS			
		č	NIGE I-	LENILARY	SEQUENCES FUR	DERIVING LOUP	3			
		C****	******	********	**********	***********	*********	************	****	
0001			IMPLIC	IT INTEGER	+2(1-N) 151.48.41.49.	NCOL NE NON'NG				
0003			CUMMON	/LOO/KH(11	1.LN(11).KS(1	1).LS(11).KT(5	) .LT(5)			
0004			COMMON	/DBL/JLOC(	20,31,JORD#(2	0,15), JROWI 20,	151			
0005			COMMON	/DBL#X/#24	15,15),M21(15	,15),MZ2(15,15	),M23(15,15	),MZ4(15,15)		
0006			COMMEN.	/PRI/11CTA	L(6)					
0007			READ(5	1) NL,NB,	NR , NCOL , NC					
8000			NK+NK-	NL .						
0010			READIS	11919) 11(KM(1)	1=1.NK)		•			
0011			REAC (5	.1.)(LM(1),	1=1,NK}					
0012	•		NS=NK-	NL						
0013			READIS	,1)(IS(1),	1=NS+NK) [=NS+NK)					
0015			READ(5	,1)(KT(8),	1=1,5)					
0016			READ(5	,17(LT(P),	1=1,5)	,				
0017			NEHS=1	= 1 . NC						
0019			NELEME	1						
0020			CALL	002	•					
0021			CALL S	PRLOC		TO 40				•
0022			WRITE!	6.3) I	-NCH-CQ-UI GU					
0024		3	FORMAT	(11,77,*	CHAIN +,13,/	1X,11(***))				
CC25		40	GOTO 4	1						
0027		5	FORMAT	(//. CHAI	N 1.13./.1X.	41(***))				
0028			GOTO 3	1						
0029		41	NCHSHN	CH				· ·		
0030			CALL A	SEQSUJ GUT RRAY	0 31					
0032			GOTO 3	2						
0033		31	ARITE (	6,8)						
0034		-	6010 2	(12, 10 10	SSIDLE CHAINS	.,				
0036		32	IFINCH	S.LT.NELEM	) GOTO 2					
0037			NCH=NC	HS						
0038		100	GOTE ( I CALL SI	100,200,30 PENCI	0;400],NELEM					
0040			GOTO 3	3						
.0041		200	CALL S	PRNG2						

# PROGRAM FOR DETERMINING TEN-LINK CHAINS WITH SPRINGS

FORTRAN	IV G LEVEL	1, MUD 4	MAIN	DATE = 70191	20/49/09		PAGE 0002	
0042		GCTC 33						
0043	300	CALL SPRNC	33					
0044		GUTO 33						
0045	400	CALL SPRAC	54					
0046	33	IF (NCH-1) 3	31.35.34					
0047	34	CALL MECCO	CM					
0048		GOTO 36			•	•		
0049	35	CALL PRINT	ſ	<i>i</i>	•	1 N N		
0050	36	NELEM=NELE	EM+1		1			
0051		IF (NELEM-4	1 32,32.2		· · ·			
0052	2	CONT INUE						
0053		NELEF=NELE	EM-1					
0054		WRITE(6,10	000)		•			
0055		00 6 1=1.4	•	1		,		
0056	6	WRITE 16.41	1,ITCIAL(I)					
0057	4	FORMATE*01	TOTAL NUMBER OF 10-L	NK CHAINS WITH ,13,	SPRING(S) - ",			
		* 15)						
0056		WRITE(6,10	000					
0059	1000	FORMAT( 1	• <b>3</b>			•		
C06C		STOP						
0061		END						

FORTRAN	IV	G	LEVEL	1,	POD 4	S	PRLEC	DATE =	70191	20/49/0	9	PAGE (	1001	
0001 CCC2 0003				SU 1# CD	BROUT II PLICIT MMONZAI	NE SPRLUC INTEGER#2(I- LL/M(15,15I,N	N) B,NL,NR,NCOL,NI	NCH NEL	#+NSTORE					
0004				15	MPEN/SI PRNG=1	PL/LPAIR (20+2	],LROW(20,15),L	.INK2(10)						
0006				Dù	1 1=1	, NB								
CCC7			1	11	KK2{1}	=0								
0000				LP	AIR(J	• • 1]=0								•
0010			-	DC	2 1=1	, NR								
0011			c 2	LR	UW(J,I	1=0								
			C SEA C	KCH	FORA	DJACENT BINAR	Y LINKS						,	
0012				00	10 1=	I ,NB								
0013				DC	120 J=	11/100-11 001 1.NL	n to							
CC15				NE	=P(J,N	CCLI								
0016				00	20 K≓	1,NE 1-18/1-x3/100	10100-50-11-001	A A A						
0018			- 20	co	NTINUE		1-1001E4111 001							
0019				GO	TO 10								ł.	
0021				16	(#13)K. [[N[]]]	//100.24.17 G K)~9000)/100.	EQ.13 GUTO 60			•				
0022			_	GC	TC 10									
			C C L IN	к 1	15 A I	BINARY LINK.	CHECK ADJACEN	I LINKS.						
0023			ັ 50	1F	EMEJ,K	+1)/100.EC.1)	GUTO 7C							
0024				16	(K.EC.	11 GOTO 51	COTC 21							
0026				GO	TO 10	-17/100+64+17								
0027			51	IF	(H(J,N	E)/100.EQ.1)	GOTO 72	· •						
0028			60	- GQ - 1 F	10 10 10 10	K+11~90001/10	0.FO.L1 GUTO 7	D						
0030				İF	K.EG.	1) GCTO 61		-						
0031				IF	([M(]+)	x-11-90001/10	0.EQ.11 GOTO 7	L.						
0033			61	IF	6626J.	NEJ-90001/100	.EC.1) GOTO 72						•	
0034				GO	TO 10									
			C LIN	ĸI	IS AD	JACENT TO AND	THER BINARY LI	NK .						
0035			L 70	LP	AIRCIS	PRNG,11=M(J,K	<b>j</b>							
0036				LP	AIRLIS	PRNG,23=#(J.K	+1)							•
0037				1.1	M(J;K+ NK2(L)	1)-(M(J)K+1)/ =1	1001+100							
0039			_	GO	TO 80									
0040			71	1.P	AIRCIS	PRNG,1]=M{J,K PRNG,21=P{J.K	1 -11							
0042				L=	M(J.K-	1)-(M(J,K-11/	1001+100					•		
0043				11	NK2(L)	-1								
0045			72	LP	AIRTIS	PHNG,]]=P[J,K	1		•			•		
0046				LP	AIRCIS	PRNG,23=P[J,N	E)							
0047				11	NK2(L)	]−(M(J;NE)/10 =1	0]+100							
			c											
									2 T 1.					
FCKTRAN	1 V	G	LEVEL	1.	MC0 4	s	PRLOC	DATE -	70191	20/49/0	9	PAGE	0002	
			с . LOC С	A TE	RONS	CONTAINING AD	JACENT BINARY	LINKS.						
0049			80	00	81 14	=1,NR								
0050				NE IF	-M(1A) (NE_E0	.01 GOTO 81								
0052				DO	82 18	=1,NE								
0053			82	1F 60	INCLA.	IB).EQ.LPAIR(	ISPRNG, []) LRO	WEISPRNG.	1#}=1A					

0053		IFIMCIA; LDJ=EQaLPAIK()
0054	82	CONTINUE
0055	81	CENTINUE
0056		I SPRNG= I SPRNG+1
0057	10	CONTINUE
0058		NCH=ISPRNG=1
0059		RETURN
6666		END

FORTRAN	IV G 4	EVEL.	L, MGD 4 CHAIN	DATE = 70191	20/49/09	÷	PAGE O	001
0001			SUBROUTINE CHAIN IMPLICIT INTEGER+2(I-N)					a de la
0003			COMMON/ALL/M(15,15),N8,NL,NR,	NCOL + NK + NCH + NELEN + NSTORE				
COCS			CCMPON/CHN/108L,KSTCRE,KGUNT	EAP(100,0)	÷			
0006	ι.		DIMENSION ANUM(10), ARRAY(8,12 DATA C/**/.D1/*-*/.C/*.*/.8	) /+i+/				
0008			DATA ANUN/101,11,121,131,141	. 5 6 7 8 9 . / . BLANK/	• •/			
0010		L	FORMAT(T5, CHAINS *,8(A1,4X,1	,D 2,6X),Al)				
0011		2	WRITE(6,2)(D,D,D,D,D,D,D,I,I=K	STORE, IDULI				
013		-		•				
015			ISUN=0					
016			1=[+] 00 11 J#1-NELEM					
010			IF(JSTORE(1A,J).GE.10) GOTO 1	2				
20			GOTO 11	N.2				
121		12	ESUP=1SUF+3 CONTINUE	- 				
23			IT=(12-ISUN)/2					
24		13	DG 13 J=1,IT Array(1,J}=Blank					
26			17=17+1 00 14 1-1 NELEM					
28			NUH=JSTORE(IA;J)					
29			NDIG=2 IE(NUM.LT.10) NDIG=1					
31			IF(NDIG.EC.2) GOTO 15					
12			ARRAY[[;[T]=ANUM[NUM+1]] TT=IT+1					
34			IF(J.EQ.NELEM) GOTO 14					
36		15	NUM1=NUM/10					
37			ARRAY[[]]]]=ANU#(NUM1+1)					
9			IT=IT+1			*		
40 61			ARRAY(I,IT)=ANUM(NUM2+1) IT=IT+1					
i		• •	IF(J.EC.NELEP) GCTO 14					
4 .		10	AKKAY[ ,[]]=L []=1]+1					
5		14	CONTINUE	and the second				
7		17	ARRAY (1, J)=BLANK	· · ·				
8		10	CONTINUE KOUNT=KOUNT-1					
•			WRITE(6,3)(B,(ARRAY(1,J),J=1,	121, I=1, KOUNT 1, B				
52		3	KCUNT#1	1,771				
53 54			KSTORE=1DBL+1 Return					
5	14 .	EVCI	END A Sports	DATE - TOLOT	2040100	•	DAGE -	
( KAN	14 0 1	EVEL		UAIE - 70191	20/44/04		PAGE (	JUQI
02			IMPLICIT INTEGER+211-N	AL, ISPANGI				
03 04			CCNHON/SPL/LPAIR(20,2),LROW(2 DIMENSION NY(15,20),M7(15,20)	0,15),LINK2(10)				
15			00 1 1=1,NR					
07			IF(NE+EQ+C) NE=1					
08 CS		,	00 1 J=NE,NCOL MZ(1.J)=0					
10		. •	L1=LPAIR(ISPRNG,1)	<b>1</b> .				
LL.	c	:	LZ=LPAIR(ISPRNG,2)					
	ġ	LOC/	TE LINK LI					
12		-	OC 10 1=1.0R					
13			NE=MY(I,NCOL) IF(NE.EQ.0) GOTO 10					
15			IF(LRCW(ISPRNG, 1). NE. T) GOTO	9 .				
			IF(MY(I,J).EQ.L1.OR.MY(I,J).E	Q.L2) GUTO 30				
17		20	CONTINUE IFINYLL.J.L. EQ. 2.08. NYLL.J.	1)-FC-(1) 6010 40				
17		~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~						
7 8 9			GOTU SU					
7 8 9 C 1 2		40	GATU 50 ICOL#1 JS#J					
17 18 19 20 21 22 23		40	GDTU 50 IGOL=1 JS=J M2(I,IGOL)=1111 I=142					
789012345		40	GD10 50 IGOL=1 JS=J MZ(1,ICOL)=1111 J=J+2 If(J.GT_NE)J=J-NE					
17 18 19 21 22 23 24 25 27		40	GIIU 50 ICOL=1 JS=J M2(1,ICOL)=1111 J=J+2 If(J.GT.NE)J=J-NE ICOL=1COL+1 M2(1,ICOL)=MY(1.J)					
7890123456780		40	GIIU 50 ICOL=1 JS=J M2(1,ICOL)=1111 J=J+2 IF[J.GT.NE]J=J-NE ICOL=ICOL+1 M2(1,ICOL)=MY(1,J) J=J+1 J=J+1					
)17 )18 )19 )20 )21 )22 )23 )24 )25 )24 )25 )26 )27 )28 )29 )30		40	GIIU 50 ICOL=1 JS=J M2(1,ICOL)=1111 J=J+2 IF(J.GT.NE)J=J-NE ICOL=ICOL+1 M2(1,ICOL)=MY(1,J) J=J+1 IF(J.GT.NE)J=1 IF(J.GT.NE)J=1 IF(J.GQ.JS) GOTO 70					
)17 )18 )19 )20 )21 )22 )23 )23		40	G(10)50 ICOL=1 JS=J M2(1,ICOL)=1111 J=J+2 IF(J=G1.NE)J=J=NE ICOL=1COL+1 M2(1,ICOL)=MY(1,J) J=J+1 IF(J=Q1.S) GOTO 70 GOTO 41 I=ME					
017 018 019 020 021 022 022 022 022 022 022 022 022		40 41 50	GIIU 50 ICOL=1 JS=J JS=J IF(J.GCL)=JLII IF(J.GT.NE)J=J=NE ICOL=1COL+1 M2(I, ICOL)=MY(I,J) J=J+1 IF(J.GT.NE)J=1 IF(J.G.JS) GOTO 70 GOTO 41 J=NE GCTO 40					
017 018 019 0221 0223 0224 0226 0226 0226 0228 0226 0228 0228 0228		40 41 50 70	GITU 50 GITU 50 JS=J JS=J JF(J,GC)=1] IF(J,GC)=F IF(J,GC)=F IF(J,GC)=F IF(J,GC)=F IF(J,GC)=F GOTO 40 J=ME GCTO 40 MZ(I,NCO)=NE-1 GOTO 10 GOTO					
017 018 0020 0221 0222 0221 0223 0024 0023 0024 0025 0027 028 029 030 031 032 033 033 035 036		40 -41 50 70 9	GITU 50 GITU 50 JS=J JS=J JF(J.GT.NE)J=J=NE IGCL=IGCL+1 M2(I,ICDL)=MY(I,J) J=J+1 IF(J.GT.NE)J=1 IF(J.EQ.JS) GUTU 70 GUTU 70 GUT					
017 019 022 022 022 022 022 022 022 022 022 02		40 41 50 70 9 8	GITU 50 GITU 50 JS=J JS=J JC1, FC0L)=1111 J=+2 IF(J,GT,NE)J=J=NE IC0L=IC0L+1 MZ(I, IC0L)=MY(I,J) J=+41 IF(J,GT,NE)J=1 IF(J,EQ,JS) GTU 70 GTU 41 J=ME GCTU 40 MZ(I,NC0L)=NE=1 GUTU 10 GUTU 10 GUT					
178921222222222223312334567890	·	40 41 50 70 9 8 10	GITU 50 GITU 50 JS=J JS=J JCI, FCOL)=1111 J=J+2 IF(J,GT,NE)J=J=NE ICOL=ICOL+1 MZ(I,ICOL)=MY(1,J) J=J+1 IF(J,EQ,JS) GOTO 70 GOTO 41 J=ME GOTO 40 MZ(1,NCOL)=NE-1 GOTO 10 D0 8 J=1,NE MZ(1,J)=PY(1,J) MZ(1,NCOL)=NE CONTINUE BFTURN					

FORTRAN	IV	G	LEV	EL	1. MCO 4	5PRNG2	OATE .	70191	20/49/09	PAGE	0001
0001					SUBBRILTINE SPR	NG2					
0002					INDI ICIT INTEG	F##2/1-N1					
0003					COMMON/ALL/ME1	5+15) +NB+NL +NR + ACCI	. NK . NCH . NEL	EM-NSTORE			
0004					COMMUN/OBALL/J	STURE(100,6), JTEMP	100,6)				
0005					COMMON/DELMX/H	Z(15,15),MZ1(15,15) (20,15,15)	#Z2(15,15)	MZ3(15,15)	,MZ4(15,15)		
0006					COMMUN/SPL/LPA	IR (20, 2), LROW( 2C, 1	),LINK2(1C)				
0007					COMMUN/CHN/ISP	RNG,KSTORE,KCUNT					
0008					1SPRNG=1						
0009					KCUNT=1						
0010					ASIUKE=1						
0012				1	FORMAT(/. OIDE	NTIFICATION OF CHAI	NS WITH TWO	SPR INGS + 1/	. LX.		
					41(***),/)						
0013					DO 100 I=1,NST	ORE					
0014			1	00	JTEMP(1,1) = JST	CRE(1,1)					
0015				~	00 10 12#1,NST	UKE					
0018					TEALL SODADOING	-NCGL . N . N 71 . F1					
0018					[P]=[+]	14602 FF (F21 117					
0019					IF[IP1.GT.NCH]	GOTC 10			-		
0020					DO 30 J+191,NC	н					
0021					JSTORE( ISPRNG,	13-1					
0022					JSTORE(ISPRNG,	2}=J					
0023					KOUNT=KOUNT+I						
0024					CALL SPRADDIAR	NCRL CRAIN					
0017			C F	111	MX	,					
0026					IROW=1						
0027					DO 20 1A=1,NR		_				
0028					LF(LRGW(1,1A).	EQ.O.AND.LROW[J][A	.EQ.0) GOTO	20	. *		
0025				۵۵	DU 4U ID=I;NCL My/ISPANG.IKOL	L ,[81=#7/[4,]8]	.,				
0031					IRCH=1ROH+1	101-111141101					
0032				20	CONTINUE						
0033		. •			MX ( ISPRNG, L, NC	OL-1 = IKOW					
0034					ISPRNG=ISPRNG+	1					
0035				30	CONTINUE						
0037				10	ISPRNG#1 SPRNG-	1					
0038					IFI ISPRNG.GE.K	STORE) CALL CHAIN					
0635					NCH=LSPRNG						
0040					RETURN						
0041					END						
COD TO AN	11/	~	1 64		1 160 4	SADNC 1	0476	. 10101	20/69/09	Parif	0881
FURIKAN		Ű				, SPANOI	DATE	10171	20/1//07		
0001					SUBROUTINE SPR	NG1 E8#2/1+81				•	
0002					COMMON/ALL/MEL	5.15).NB.NI.NR.NCO	NK .NCH .NEI	EN-NSTORE			
0004					COMMON/SPL/LPA	1R(20.2).LHEW(20.1	51.L [NK2110]				
0005					COMMUN/OBALL/J	STURE ( 100,6), JTEPP	100,61				
0006					COMMUN/08LNX/M	2(15,15),M21(15,15) (20.15.15)	MZ2(15,15)	M23(15,15)	,MZ4(15,15)		
0007					WRITE(6,1)						
8000				1	FORMAT(/, OCHA	INS WITH ONE SPRING	G LIDENT IFI	ED BY THE SP	RING NUMBER		
0009					00 10 1=1,NCH						
C010					I SPRNG=1						
0011					$JSTORE{[,1]=I}$						
0012					CALL SPRADDINR	,NCOL, F, MZ, I)					
0013			L F		.⊓∧  8C₩#						
0014					DO 20 LA=1.NR						
0015					IFILROW(1, IA).	EQ.0) GOTO 20					
0016					DD 40 18=1,NCO	L					
0017				40	MXLISPRNG, IRUN	,18)=HZ(1A,18)					
0018					INDW= INDW+1						
0015				20	LUNIINUE	()					
0021				10	CONTINUE	0L-1/-1808					
0022					NCH*ISPRNG				4		
0023					RETURN						
0024					END						

FORTRAN	IV-G LEVEL	1, MOD 4	SPRNG 3	DATE	# 70191	20/49/09	PAGE	0001
0001		SUBROUTINE SPI	INGB					
0002		IMPLICIT INTER	SER#2(1-N)					
0003	1	COMMON/ALL/MI	5,151,NB,NL,NR,NC	JL .NK .NCH .NE	LEM.NSTORE	1		
COC4		COMMON/DBALL/	STCRELLOO.6). JTEM	P(100.6)	•			
0005		COMMON/DBLMX/I	2(15,15).#21(19,1)	51.M22115.15	1.HZ3(15.1	5),MZ4(15,15)		
		*.H25(15.15).H	(20.15.15)		••••••			
COCE		CCPPON/SPL/LP	1R(20.2).LKC.(20.	15).LINK2(10	3			
0007		COMMON/CHN/LSI	RNG .KSTORE .KOUNT					
0008		ISPRNG=1	•••••					
0009		KOUNT=1						
0010		KSTORE#1						
0011		WRITE (6.1)						
0012	1	FORMAT(/. +OLD	INTIFICATION OF CH	LINS WITH TH	REE SPRING	51./.18.		
•••••	•	*43(*+*)./)						
0013		00 100 L#1-NS	[C86					
0014		JTEMP(1.21=15)	IORE(1.2)					
0015	100	ITE #041.11=.151	CREELLIN					
0016		00-10 17=1-05	ORE					
0017		16117-60-11 6	10.9					
0018		IFA ITCMP117.1	.NE. ITEMP(17-1.11	6 010 9				
0019		6010 8				•.		
0020	9	1#JTENP(17.1)				·		
0021		CALL SPRADDIN	L.NCOL.N.N71.11					
0022	A	Ja ITEMPI 17.21	(Jacob Je Jezz Ji J					
0023		CALL SPRADOIN						
0024		ID I + IAI	(heele thet heel of					
0025		ICLIDE CT NCH	0100					
0025		00 30 Kz 101 M						
0027		ISTORF/ISPRNG	.1) = 1					
0028		ISTORETICONC	71=1					
0020		ISTORELISPANO	31=5					
0010		KOUNT-KOUNT+1	131-K					
0030		ICINGINE CE A		· 、				
0032		CALL SDDADDIN	- NCD1 - N72 - N7 - K1					
0034	-C - E 11	CALL SPRADUIN	(Incue the status ins					
	•U F IL	1906-1	1 A					
0034		00 30 1Å-4 NU	· ·					
0034		LET DOWLE TANK	EQ O AND LOOKES. L.	I ED A MAD	POWLY TAL	50 01 0010		
0033		4 30	CALO SWIDICKOWI AT I			Lettor Guid		• •
0036		- 20 DO 40 [H+1 NC						
0030	40	WY/ICODAC TUCK	JE. 1 191-11/14 101					
0037	40	TATISPRAUJIKU	5101-MC(1A, 10)					
0030	10	IKUWFIKUWFI			•			
0039	20	CONTINUE	- 1 1 - 100H	`				
0040		TALISPENGILING						
0041		1 SPKNGTI SPKNG	r1			· · ·		
0042	30	CONTINUE				ter generative de la constance		
0043	10	LUNIINUE						
0044		ISPRNG# ISPRNG						
0045		ITTISPENG.GE.	STUKEJ CALL CHAIN					
0046		NCH= ISPRNG						
0047		RETURN						
0048		ENU				•		

F	ORT P AN	٤v	G	LE	VEL	1	, ,	00	4	,	SPRNG 4		OATE	ŧ =	70191	20/4	9/09		PAGE	0001
(	1001					s	U8F	.OU	T 1	NE SPRNO	i4									
	002					1	PPE	10	1 Ť	INTEGER	t+2(1-h)									
(	003					Ċ	GNI	ICN.	/ 8	LL/HLIS	151-NB-NL-NR-NC	DL •NK •	NCH.	NELI	EM.NST	ORE				
- 1	0064					č	DM2	INN	/1	BALLIST	CRE4100.61.JTEM	P(100.	6)							
	0005					č	0.00	<b>ON</b>	20	BI MX/M7/	15-151-071/15-1	51.872	115.1	151		5.151.074/15.	151			
						•			5.	151.MX12	0.15.151			• • •						
	10.04					72		0.	20	DI /I DATE	120.21 1066120.		*****							
	0000					2			, s	HN /TCROK	C VETORE VOUNT		NR2 11			·				
										MALI SPAN	IG & KSTUKE & KCONT									
							3 r f		-1										-	
	1004					K	101		١.									÷		
	010					ĸ	510	RE	* 1							•				
	1011					- 14	REI	IE (	6,	IJ										
	012				1	F	DRI	IA T	()	••010EN1	IFICATION OF CH	ALNS N	I TH I	FOU	R SPRI	NGS+#/#1X#				
						\$4	2()	+	1,	11										
. (	013					0	וכ	00	1	=1,NSTOP	E									
- (	0014					J	161	IP (	l,	3)=JST0#	(E11,3)									
	015					J	TE)	PL	۱.	2) = JSTOF	LE(1,2)									
	016				100	4	TEX	191	1.	1)=JSTOR	ELT.L.		1							
	017					ō	o I	0	iż	=1.NSTOP	LE .									
i	018					ī	F 4 1	2.	F۵	.1) 6010	1 9									
ć	1019					1		TE	MD	(12.11.)	E. ITEMPIT7-1.11	0100	9							
2	1020					1	51	ITE	-	117.21.1	E. ITENE(1/-1.7)	1 6010								
	1021					÷			nr							•				
	1022				•				a t		•									
	022					2	- J	10	r 1	14913										
	1023					5			r K	AUD INK I	NCUL # M # M # I I I I I									
	1029				8	4	. J	ER	26	14,23										
	1025				_	Ç	ALI	<u> </u>	PK	LADD (NH.)	ICDL ##21+#22+J}									
	3026				1	ĸ	- J I	EM	Pł	12,3)										
	1027					C	AL L	. S	PR	AOD(NR ,	NCOL, MZZ, MZ3,KI									
	2026					K	P1 -	•K+	1											
	3029					1	F ( )	(P 1	•6	T+NCHI (	GTO 10					·				
	030					D		10	L=	KP1,NCH										
	031					J	S T (	JKE	"	SPRNG , 1	1=1		•							
•	1032					J	STO	JRE	(1	SPRNG, 2	)=J									
. (	033					J	sta	INE	(1	SPANG, 3	}=K									
	0034					J	51(	RE	11	SPRNG . 4	) =L									
Ċ	035					ĸ	ÚUN	11=	κÖ	UGAT+1										
i	0036					Ĩ	FLI	ίοι	NT	-GT-8) (	CALL CHAIN									
1	037					č		S	PR	ADDINR .	COL. H23.H2.L)									
				С	E 11.	ı.	HX												, ,	
	038			•	• • •	٦,	ROI	4 = 1								1. A				
	1039					'n		0	• •	ALL NO										
	1040					ĩ		50				A1 E0		• •	. 4 1010	LAN ED. O. AND.				
	1040					.'					A COTO - 20							· ·		
						•_			58	IAJ + EQ+	0010-20									
	2041							10	10	I I NLUL										
	1042				40		A 1 1	36	KN	IG # I KUW# I	D] == ( [ A ; [ D ]							,		
9	1043					1	KU	1= I	RÜ	IM+T						1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -		1 N		
. (	1044				20	C	LNI	IP	UE											
	045					M	×( 1	SP	RN	16 <b>, 1 , NC</b> OL	1J=180W									
- 0	046					1	5 P F	ING	= i	SPRNG+1										
•	1047				30	С	CNI	I I N	UE											
	048				. 10	C	ONI	T E Ņ	UE											
•	049					1	SPI	RNG	= 1	SPRNG-1										
	0050					1	F ( 1	SP	RŅ	G.GE.KS	FORE) CALL CHAIN									
1	051					N	CH	15	PR	NG										
ì	052					R	ETL	IRN		-										
. 2	1053					F	ND									•				

۰,

FORTRAN	IV G LEVEL	1. HCD 4	1.1	PRINT		DATE =	70191	20/49/09	PAGE	0001
0001		SUBROUTINE	PRINT							
0002		IMPLICIT I	NTEGER+21	(-N)						
0003		INTEGER IF	0	• •••						
0004		COMMON/ALL	/8(15.15)	.NB .NL .NR .	NCOL . NK .	NCH-NEL	EP-NSTORE			
COCS		COMMON/PRI	ITCTALI6	)						
0006		COMMON/MEC	/IEQLIVISO	0.50)						
0007		DIMENSION	PRINTI 201	)						
0008		DATA LEQ/1	=1/							
0009		I SUBT=NCH								
0010		WRITE16,31								
0011	3	FORMAT( 'O'	15, EGUIN	VALENT CHA	AINS ./ )					
0012		IFINCH.EQ.	L) GOTO 40	0						
0013		DC 30 IX=1	NCH							
0014		ISUM#0								
0015		CO 20 IY=1	NCH							
0016		IFEIEQUIVE	[X, [Y].EQ.	.0) GCTO 2	20					
0017		ISUN=ISUN+	1							
0018		IPRINTIISU	4) = [EQU IV	(1X,1Y)						
. 0019	20	CONTINUE								
0020		IF(ISUM.EQ	.0) 6010 3	30						
0021		WRIJE(6,1)	IX.(1EQ.)	LPHINT ([]),	1=1,ISUM	3				
0022	1	FORMATITIO	,12,10(A1,	121)						
0023		ISUBI=ISUB	I-ISUM							
0024	30	CONTINUE								
0025		IF(ISUBT.E	2.NCH) GO1	FO 40						
C026	31	WRITE (6,2)	I SUBT							
0027	2	FORMAT('O'	, T5 , ' NUPBE	ER CF LNIC	QUE CHAIN	S -1,12	23			
COZB		ITCTALINEL	EM3=1TOTAL	L (NELEM)+I	ISUBT					
0029		RETURN								
0030	40	WRITE(6,4)								
0031	4	FORMAT (T10	*NONE*}							
0032		GOTO 31								
0033		END								

FERTRAN	IV G LEVEL	1, MGD 4	ARRAY	DATE = 70191	20/49/09	PAGE 0001
0001		SUBROUTINE ARR	AY			
0002		IMPLICIT INTEG	ER#2([-N]			
0003		COMMENZALL/MIL	5.151.N.8.NL .NR .NC	OL .NK .NCH.NELEM.NSTORE		
0004		COMMON/SPL/LPA	18120.21.18CH120.	5).LINK2(10)		
0005		DIMENSIUN LLIC	0.3)			
0006		QATA E/1 1/.P	L/+1+/.PR/+3+/.C/	• • • /		
0007		DO 10 TEL-NCH		•••		
0008		1 (1.1)#1 PAIR (1	.11-11 PATR (1.11/1	001+100		
0005	10	1 (1.2) #1 PA1811	-21-11 PATRIL-21/1	001+100		
0010		WRITE(6.1)				
0011	1	EORMAT ( COLDENT	LETCATION DE SPRI	142.04		
0012	•	WRITELS. 2111.1	TI NCHI			
0013	2	EORMAT/ IOI. TS.	1 SPU (NG#1. T18. 116	1 1.12.3711		
0014	•	HRITEL6.31(0.0	.D. Isl. NChl	1 JICISALI		
0015		CORMATITS. 13/1				
0016		UUITE/A.41/11	11.0 111 21 1-1.1	17 141		
0017	4	ECOMAT/TS ALLA	91790701172795-19 81. 17883 4.11741	1.12.41.12.191.775		
0011	-	DETION	NET LENKE TELET			
0010		CLO CLO				
0014		END				

FCRERAN	IV G	LEVEL	1, MOI	) 4		BLK DA	TA	DATE .	70191	20/49/0	PAGE DOG	1
0001			BLCCK	DATA								
0002			IHPL10	LIT EN	TEGER#211	-11						
0003			COMMON	VD8AL	L/JSTORE(	100,6)	JTEMP(100,6	)				
0004			сойног	VPRI/	ITOTAL (6)							
0005			CATA 1	TOTAL	/6*C/							
0006			CATA .	ISTORE	/600+0/,J	TEMP/6	00+0/					
C007			END	,								



Figure 24. Flow Chart for Subroutine SPRLOC





#### APPENDIX H

# SUBROUTINES USED IN DETERMINING THE TEN-LINK CHAINS WITH PULLEYS AND BELTS

The program for determining the chains with pulleys and belts is the same as that shown in Appendix G with Subroutine SPRLOC being replaced by Subroutine PULLOC and Subroutine SPRNG1 through SPRNG4 being replaced by PUL1 through PUL4. Also, some format statements in the output subroutines require changes.

Subroutine PULLOC is shown in Table XLIX and its flow chart in Figure 26. Subroutine PUL3, which demonstrates the logic for all PUL subroutines, appears in Table L and its flow chart is found in Figure 27.

## TABLE XLIX

## SUBROUTINE PULLOC

FORTRAN	1V G	LEVEL	1, 80	D 4	PULLOC	DATE = 70192	09/56/51
			£11000				
0001			20040				
0002			CONNO		GER #211-NJ		
0003			COMMO	4/ALL/711		.UL 14K 1464 14EL EA14310KE	
0004			LONN	N/ FUL/ JFU	OF #10441*36KOM(10)	1 7 5 1	
0005			IPUL=				
		C					
		LSEA	KCH PU	R FERNARI	T LIAKS		
		6					
0006			00 10	U K#1;NO			
0007			00 10				
0008			16446				
0004			111146	• EQ • UJ UL	10 10		
0010			UU II	J=194E		() CDTO 20	
0011			CONTR	1131-1N() MIE	1 377 1001-1001241		
0012		10	CONTI				
0014		20	IEIMI	1.11/100	NE.31 GOTO 100	*	
0014		<u>د</u> د	1	11317100	RE: 31 0010 100		
		Č DET	CONTINE	LE TERN	ABY LENK IS JOINED	TO THE RINARY LINKS	
		r	- NATUR				1. A
0015		•	00 30	TART.NR			
0016			NFaMI	A-NCOL 1			
0017			<b>LEUNE</b>	-EQ.01 G	010 30		
0018			00 40	[8=1.NE			
0019			TELME	1A.181.E	0.M([.J]) GOTO 41		
0020		40	CONT I	NUE			
0021			GOTO	30			
0022	•	41	14-18-	-1			
0023			12=18	+1			
0024			IFIIB	.EQ.11 I	Y=NE		
0025			IY1=I	Y-1			
0026			LECIV	.EQ.11 1	¥1=NE		
0027			1F(18	.EQ.NEJ I	12=1		
0028		1.1.1	121=1	2+1			
0029			IF(12	.EQ.NEJ	[2]=1		
00 30			IFINI	IA, IY)/10	00.EQ.1.AND.H(IA,	[2]/100.EQ.1] GOTO 42	
0031			GOTO	30			
0032		42	JPUL	(PUL,1)=	H(IA,IB)		
0033			JPUL	[PUL=2]=	M(IA, 1Y)		
0034			JPUL	IPUL 31*	MEIA, IVII	,	
0035			JPUL	IPUL . 41=1	M(1A+12)		
0036			JPUL	IPUL (5)=	M(1A,121)		
0037			IPUL*	IPUL+1			
0038			CONTR	43 			
0039		30	COTO	NUE			
0040		<i>c</i>	<b>6</b> 010	100		•	
		Č DET	COMINE	TE A TH	TRO BINARY & THE C	CONNECTED TO THE TEAM	ARY LINK
		č		•• •• •••			
0041		ِ 4٦	DO 60	IC=1.NR			
0042			NE=MI	IC NCOL1			
0043			IFINE	.EQ.01 G	010 60		
0044			DO 50	ID=1.NE			
0045			TECHL	IC. 10) . E	Q.M(I,J)) GUTO 61		
0046		50	CONTI	NUE			
0047			GOTO	60			
0048		61	11=10	-1			
0049			12=1D	+1			4
						-1-	

PAGE 0001

FORTRAN	V G LEVEL	1, MOD 4	PULLOC	OATE = 70192	09/56/51	PAGE 0002
0050		IF(10.EQ.1) 11	.=NE			
0051		111=11-1				
0052		IF(11.EQ.1) []	1=NE			
0053		IF(10.EQ.NE) 1	2=1			
0054		121=12+1				
0055		1F(12.EQ.NE)12	1=1			
0056		IF(M(IC,11).NE	.JPUL(1PUL-1,2).AM	ID.M(IC, I1).NE.JPUL(IPUL	1,4)}	
		1 GOTO 70				
0057		IF (M(IC,12).NE	.JPUL([PUL-1,2].AN	10.M(IC,I2).NE.JPUL(1PUL	1,4}}	
0054	40					
0050	70					
0057		15/M(10.111/10	0-NE-11 6070 100			
0061		1041 (1044.71m				
0062		IPHI (TPHI . 3) =	446.1111			
0063		JPIN (1PUL+).21	=M(IC.I))			•
0064		JPIN ( [PIIL +1 -3]	##(IC.111)			
0065		6010 72				
0066	71	JPULITPUL-1.61	=M(IC.12)			
0067	• -	1FIMILC.121/10	0.NE.11 GOTO 100		· _	
0068		JPUL(1PUL+2)=	([C.12]			
0069		JPUL(1PUL.3)=P	(16.121)			
0070		JPUL(1PUL+1+2)	=M(IC,12)	· ·		
0071		JPUL ( IPUL +1, 3)	=M(1C,121)			
0072	72	JPUL(IPUL,1)=	([,])			
0073		JPUL(IPUL,4)=J	IPUL(1PUL-1,2)			
0074		JPUL(IPUL,5)=J	IPUL(IPUL-1,3)			
0075		JPUL(IPUL,6)=J	PUL([PUL-1,4]			
0076		1PUL=[PUL+1				
0077		JPUL(IPUL:1)=	1([,J]			
0076		JPUL(1PUL,4)=J	PUL(1PUL-2,4)			
0079		JPUL (19UL , 5)=J	PUL(IPUL-1,5)			
0080		JPUL(1PUL:6)=	PUL(1PUL+2;2)	1		
0081						
0082	100	LUNIINUE				
0083	~	ACH-IPOL-1				
	C DET	ERMINE ROWS CON	ITAINING PULLEY			
	C					
0084		IF{NCH.EQ.0} P	RETURN			
0085		00 110 1=1,NCH	1			
0086		DO 110 J=1+NR				
0087		JPROW(1,J)=0				
0088		NE=M(J; NCOL)				
0089		(F(NE.EQ.0) GC	110 110			
0090		DO 120 K=1.NE				
0091		LFCM(J.K).EQ.J	PUL(1,11) GOTO 121			
0692	120	CONT INUE				
0093		6010 110				
0094	121	JPRUWII,JJ#J				
0042	110					
0096						
0047		ENU				



Figure 26. Flow Chart for Subroutine PULLOC

#### TABLE L

### SUBROUTINE PUL3

FORTRAN IV	G LEVEL	1, MOO	14	PUL 3	DATE = 70192	09/56/51	PAGE 0001
0001		SUBBOU	TINE PU	11.3	·		
0002		IMPL 1C	IT INTE	GFR#2(1-N)			
0003		COMMON	/ AL 1 / MI	15.151.NB.NL.NR.NCO	L . NK . NCH . NEL EM . NSTORE		
0005		COMMON	/DBALL	ISTORE (100.41. ITEMP	(100.6)		
0004		COMMON	VOALLY	W2/15.151.W71/16.15	1.072/15.151.073/15.151	- #24(15-15)	
0005				MC(13)1371021113913	11022(1)11)1 (RES(1)11)	102 4(1 )11 )1	
		* ****	1241214	MA(13)13)13)	E 1		
0006		CUMMUN	17 P UL 7 JP	UL(10,6), JPROW(10,1	21		
0007		CUMMUN	/CHN/IP	UL,KSIDKE,KUUNI			
0008		COMMON	I/LIN/LI	NE, IPAGE			
0009		DIMENS	TON LIG	5),L2(5),L3(5),L4(5	)		
0010		1PUL≠1					
0011		KOUNT=	1				
0012		KSTORE	=1				
0013		IFILIN	1E.GT.47	') CALL PAGE			
0014		WRITE(	6,1)				
0015	1	FORMAT	1/, 1010	ENTIFICATION OF CHA	INS WITH THREE PULLEYS"	,/,1X,	
		* 43(**	11./)				
0016		LINE=L	INE+5				
0017		DO 100	[=1.NS	TORE		1	
0018		JTEMPO	I-21=J5	TORE(1.2)			
0019	100	JTEMPO	1.1)=15	TORE(1,1)			
0020		00 10	12=1.05	TORE			
0021		IE117.	F0.11 0	9 010			
0022		TELATE	MP(17.1	).NE.JTEMP(12-1.1))	6010 9		
0023		6010 8					
0024	q	La.ITEM	PL 17.11				
0025		00.91	IAs1.5				
0026	91	11/14	#.IPIII ( 1				
0027	~	CALL		NR-NCOL-M-M71)			
0028	9	IN ITEN	D/17.71				
0020	9	00 01	TA=1.5				
0020	91	121341					
0030		CALLO		NR.NCOL.M71.M721	1. L.		
0032		101 = 14	1				
0032		16(10)	GT NCH	0 6070 10			
0034		00 30	N= 101.N				
0034		00 71	14-1.5				
0033	71	13/141	- IDIII 4 M				
0030	•1	16/114	11. 60.1	3(1)1 6010 30			
0034		16/11/	2) 50.1	3/31.00.11/21.50.13	(51) COTO 30		
0038		15111	A1 E( 1	3/3) 08.11/41.60.13	(5)) COTD 30		
0039		1671.37	11 50 1	3(1)) 6010 30			
0040		161121	21 60 1	2121 00 12/21 50 12	(81) 6010 30		
0041		12/1 74	41 60 4	3131.00 13441 CO 13	(51) 6010 30		
0042		101020	41+CQ+L	.313/+UK+L214/+E4+L3	())) 0010 30		
0043		JUNE	CIPUL 1				
0044		JSTURE	TIPUL 2	(7=J			
0045		JSTURE	I I PUL 3	) <b>;</b> = K			
0046		KOUNT-	KOUNT+1				
0047		IFEKDU	INT.GT.5	CALL CHAINI			
0048		CALL P	ULADDIK	NKINCULIMZ2IMZ)			
	C FIL	LMX					
0049		1R0W#1					
0050		00 20	1A=1,NR				
0051		IF(JPR	IOW(1,1/	I.NE.IA.AND.JPROW(J	, JAJ.NE. IA.AND. JPROW (K,	(A).NE.IA)	
		* GOTO	20				
0052		DO 40	18=1,NC	OL			
0053	40	MX(IPL	JL, IROW,	IB)=HZ(IA,IB)			
0054		[ROW=1	ROW+1				
					н. Т		
				o.u. 0			
PURIKAN IV	O LEVEL	I MUL	. 4	PULS	DAIE = 10145	04/20/21	PAGE 0002
0055	20	CONTIN	41F				

.

 0055
 20
 CONTINUE

 0056
 MX(IPUL\_1I,RCOL\_1)=IRGW

 0057
 IPUL=IPUL+1

 0058
 30
 CONTINUE

 0059
 10
 CONTINUE

 0060
 IPUL=IPUL-1

 0061
 IF(IPUL\_GE\_KSTORE)

 0062
 NCH=IPUL

 0063
 NSTORE=NCH

 0064
 RETURN

 0065
 END

.

.

•

FORTRAN	IV G	LEVEL	1, HOC	4	PULADD	DATE = 70192	14/06/25	PAGE 0001
0001			SUBROU	JTINE P	ULADD( I, NR. NCGL, MY, MZ)			
0002			I HPL IC	IT INT	EGER+2(I-N)			
0003			COMMON	VPUL/J	PUL(10,6), JPROW(10,15)			
0004			DIMENS	ION MY	(15,20),MZ(15,20)			
0005			L1=JPU	11(1,1)				
0006			L2=JPL	JL{1,2}				
0007			L3=JPL	JL(1,4)				
0008			DO 10	J=1,NR				
0009			IF(JPA	(Ow ( I , J	).NE.J) GOTO 2			
0010			NE=MY (	J. NCOL	)			
0011			DQ 20	K=1, NE				
0012			IFUNYI	(J.K).E	Q.L11 GOTO 30			
0013		20	CONTIN	IUE				
0014		30	I STOP-	rK.				
0015			ICOL=1					
0016			MZ{J <sub>1</sub>	[]=1111				•
0017		31	ICOL=	COL+1				
0018			K=K+1		•			
0019			IF(K.	it.NEJK	=			
0020			IF{Kal	Q. 1510	PI GUTU 3			
0021			M21J,		71J963 0 1 9 00 Mull VI FO 1 91	N141 100 1-1211		
0022			IPUNT	Jakise	Q. L 2. UK. MY (J. KJ. EW. L 3)	MELITICUL F=1211		
0023		•	GUIU	31				
0024		5	K=NE+	L 1	01			
0025				(K=K ; NC				
0026		•	HZLJ,		Jekkj			
0027					•			
0028			00 31	(*************************************	L			
0029		10	CONTIN		4 N 2			
0030		10	DETUR					
0031			END	•				
<u>9932</u>			END					



Figure 27. Flow Chart for Subroutine PUL3

..

### APPENDIX I

# SUBROUTINES USED IN DETERMINING THE TEN-LINK

### CHAINS WITH CAM PAIRS

As with the program for determining chains with pulleys (Appendix H), this program follows the basic form shown in Appendix G. Subroutine CAMLOC, which locates possible cam pairs, appears in Table LI, and its flow chart in Figure 28. Subroutine CAM3, which develops the chains with 3 cam pairs, is shown in Table LII and its flow chart in Figure 29.

## TABLE LI

## SUBROUTINE CAMLOC

FORTRAN LY	G LEVEL	1, MCD 4	CANADD	DATE = 70193	14/26/56	PAGE 0001
0001			D(K.NR.NCOL.MY.N7)			
0001		JUDRUUTINE CARAC	UTKING NEULINTINZI			
LUCZ		IPPLICIT INTEGER	(+2(1-N)			
0003		COMMON/CAL/JCAMI	8,3),JCRON(8,15)			
0004		DIMENSION MY(15.	15).MZ(15.15)			
0005		LI#JCAN(K-1)				
0006		00 20 (m) NO				
0000		Server and the server				
0007		IFI JUNUWIN, IJONE	-1} 6010 25			
3000		NE#MY(1,NCOL)				
0009		DO 30 J=1,NE				
CO10		1F(MY(1.J).EC.L1	) GUTO 40			
0011	30	CONTINUE				
0012	40	ISTOP=J				
0013		1001=1				
0014		100L-1				
0014		M261, ICUL J=1111				
0015	41	1000=1000+1				
0016		1=1+1				
0017		IF(J.GT.NE)J=1				
0018		IF(J.EQ.ISTOP) G	010 42		-	
0019		H741.1COL)=HY11.				
0020		6010 41	•••			
0021	43					
0021	42	OO FO LO LAGOL				
0022	43	DU SC L=J,NCUL				
0023	50	MZ{[;L}=MY{[;L}				
CC24		GOTO 20				
0025	25	J=1				
C026		GOTC 43				
0027	20	CONTINUE				
0027	20	CONTINUE				
0028		RETORN				
0025		END				
				•		
FORTRAN LV	G LEVEL	1. MOD 4	CANLUC	DATE = 70193	14/26/56	PAGE 0001
0001		SUBDOUTTNE CANLO	r			
0001		SUBROUTINE CARLO				
6662		IMPLICIT INTEGER	+211-N3			
0003		COMMON/ALL/M(15,	15), NB, NL, NR, NCOL, N	K,NCH,NELEM,NSTORE		
0004		COMMON/CAL/JCAM(	8,3),JCROW(8,15)			
0000		1048-1				
0005		ILAN-I				
0005	c	ILAM=1				
0005	C SEA		MKS			
0005	C C SEAI	RCH FOR BINARY LI	NKS			
0005	C C SEAI C	RCH FOR BINARY LI	NKS			
0006	C C SEAI C	ICAN-I RCH FOR BINARY LI DO 100 K=1,NB	NKS			
0006	C C SEAI C	ICAN-I RCH FOR BINARY LI DO 100 K=1.NB DO 10 I=1.NL	NKS			
0006 C007 0008	C C SEAI C	ICAN-I RCH FOR BINARY LI DO 100 K=1,NB DO 10 I=1,NL JCRUMIICAM,I}=0	NKS			
0006 C007 0008 CC09	C C SEAI C	RCH FOR BINARY LI DO 100 K=1,NB DO 10 I=1,NL JCRUWIICAP,I}=0 NE=P(I],NCCL}	NKS			
0006 C007 0008 CC09 0010	C C SEAI C	TCAH-1 RCH FOR BINARY LI DO 100 K=1,NB DO 10 I=1,NL JCRUWIICAM,I}=0 NE=P(1,NCCL) DO 10 J=1,NE	NKS			
0006 C007 0008 CC09 0010	C SEAI C SEAI C	CCAP-1 RCH FOR BINARY LI DO 100 K=1,NB DO 10 I=1,NL JGRUWIICAM,I)=0 NE=M(1,NCCL) DO 10 J=1,NE IE(N(1,1)=(N(1,1))	NKS	10 20		
0005 0006 0008 0008 0010 0011	C SEAI C SEAI	CCAP-1 CC FOR BINARY LI CC 100 K=1,NB DO 10 I=1,NL JCROW(ICAP,I)=0 NE=P(I,NCCL) DO 10 J=1,NE IF(M(I,J)=(M(1,J))	NKS ]/100}*100.EQ.K} GU	ŦU 20		
0005 0007 0008 0009 0010 0011 0012	C 5EAI C 5EAI	RCH FOR BINARY LI 00 100 K=1,NB 00 10 I=1,NL JCRUHICAP,IJ=0 NE=#(1,NCCL) 00 10 J=1,NE IF(MII,J)=(MI1,J) CONTINUE	NKS 1/100)+100.EQ.K} GU	TU 20		
0005 0006 0008 0008 0010 0011 0012 0013	C 5EAI C 5EAI 10 20	CCAP-1 CC FOR BINARY LI CC 100 K=1,NL DC 101 I=1,NL JCROW(ICAP,I)=0 NE=P(I,NCCL) DO 10 J=1,NE IF(N(I,J)-(N(I,J) CCNT1NUE IF(N(I,J)/100-NE	NKS )/100)+100.EQ.K) GU .1) GCTC 100	TU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014	C 5EAI C 5EAI 10 20	CCAP-1 CC FOR BINARY LI CC 100 K=1,NB D0 10 I=1,NL JCCWHICAF,13=0 NE=#(1,NCCL) D0 10 J=1,NE IF(MI,J)=(M(1,J) CCNTINUE IF(MI,J)=(100,NE JCAM(1CAM,1)=M(1)	NKS 1/100)+100.EQ.K} GU .1) GCTC 100 .J	TU 20		
0005 0006 0008 0008 0010 0011 0012 0013 0014 0015	C 5EAI C 5EAI 10 20	CCAP-1 CCAP-1 CC 100 K=1,NB D0 10 I=1,NL JCRUWIICAM,I)=0 NE=P(II,NCCL) D0 10 J=1,NE IF(M(I,J)-(M(I,J) CCNTINUE IF(M(I,J)/100.NE JCAM-11AM,I)=M(I IX=J01	NKS 1/100]+100.EQ.K) GU .1) GCTC 100 .J)	TU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014 0015 0016	C SEAI C 10 20	10.4-1         00       100       K=1,NB         00       10       i=1,NL         00       10       i=1,NL         00       10       j=1,NL         10       10       i=1,NL         10       10       j=1,NL         16       11,J-1(N(1,J)       i=1,NL         16       M(1,J)-1(N(1,J)       i=1,NL         17       J=1       i=1,NL	NKS ]/100]+100.EQ.K} GU .1] GCTC 100 .J]	TU 20		
0005 0006 0008 0008 0011 0012 0013 0014 0015 0016 0017	C SEAI C 10 20	ICAR-1         RCH FOR BINARY LI         D0 10 1=1,NL         JCRUMIICAM,1]=0         NE=P(1,NCCL)         D0 10 J=1,NE         IF(MI,J)-(MIL,J)         CCNTINUE         IF(MI,J)-(MIL,J)-(MIL,J)         CCATINUE         JCAN(1CAM,1)=M(1)         IX=J+1         IY=J-1         IF(J=EQ.NE)IX=1	NKS 1/1003+100.EQ.K) GU .1) GCTC 100 .J)	TU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014 0015 0016	C 5EAI C 10 20	10.4-1         00       100       K=1,NB         00       10       i=1,NL         00       10       i=1,NL         00       10       i=1,NL         00       10       i=1,NL         01       0       i=1,NL         01       0       i=1,NL         01       0       i=1,NL         10       10       i=1,NL         15       i=1,NL       i=1,NL         16       i=1,NL       i=1,NL         16       i=1,NL       i=1,NL         17       i=1,NL       i=1,NL         16       i=1,NL       i=1,NL         17       i=1,NL       i=1,NL         16       i=1,NL       i=1,NL         17       i=1,NL       i=1,NL         18       i=1,NL       i=1,NL         17       i=1,NL       i=1,NL         17       i=1,NL       i=1,NL         17       i=1,NL       i=1,NL         17       i=1,NL       i=1,NL         16       i=0,L       i=1,NL         16       i=0,L       i=1,NL	NKS ]/100]*100.EQ.K) GU .1) GCTC 100 .J)	TU 20		
0005 0005 0008 0008 0011 0012 0013 0014 0015 0016 0017 0019	C SEAI C 10 20	10.4-1         CCH FOR BINARY LI         CO 10.4-1,NB         D0 10.4-1,NL         JCRUHILGAF,13=0         NE=#(1,NCCL)         D0 10.3-1,NE         IF(MI,3)-(MI1,3)         CONTINUE         JF(MI,3)-(MI1,3)         CONTINUE         JCANILGAN,13=M(I         IX=3-1         IF(J_EQ_NE)IX=1         IF(J_EQ_NE)IX=1         IF(J_EQ_NE)IX=NE         JCAMILGAN_23=#10	NKS 1/100)*100.EQ.K} GU .1) GCTC 100 .J)	TU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014 0015 0016 0017 0016 0019 0019	C 5EAI C 10	10.4-1         00       100       K=1,NB         00       10       i=1,NL         00       10       i=1,NL         00       10       i=1,NL         00       10       i=1,NL         01       0       11,NL         01       0       11,NL         10       10       i=1,NL         10       10       i=1,NL         10       10       i=1,NL         10       10       i=1,NL         11       11,NL       i=1,NL         12       11       i=1,NL         14       1,J/100.NE       jCAN4(1CAN, 1)+M(1,J)         14       J=1       i=1         17       J=1       i=1         17       J=0       N=NE         14       J=1       i=1         15       J=0       N=NE         J=1       I       J=2         J=1       I       J=3         J=2       J=3       J=3     <	NKS 1/100)+100.EQ.K) GU .1) GCTC 100 .J)	TU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014 0015 0016 0017 0016 0017 0019 0020	C SEAI	ICAM-1         RCH FOR BINARY LI         D0 10 1=1,NL         JCRUHILGAM,13=0         NE=#(1,NCCL)         D0 10 J=1,NE         IF(MI,J)=(N(1,J)         CONTINUE         IF(MI,J)=(N(1,J)         LGANICAM,13=M(1         IX=J=1         IF(J,EQ,NE)IX=1         IF(J,EQ,NE)IX=1         JCAMICAM,23=M(1         JCAMICAM,23=M(1	NKS 1/100)+100.EQ.K) GU .1) GCTC 100 .J) .X)	TU 20		
0005 0005 0007 0008 0010 0011 0012 0013 0014 0015 0016 0017 0016 0017 0016 0019 0020	C SEAN C SEAN 10 20 C DETI	ICAM-1         RCH FOR BINARY LI         D0 10 1=1,NL         JCRUHILCAM, 1)=0         NE=P(1,NCCL)         D0 10 J=1,NE         IF(MI,J)=(N(1,J)         CONTINUE         IF(MI,J)=(N(1,J)         CONTINUE         JCANHILCAM, 1)=NL         IF(MI,J)=(N(1,J)         CONTINUE         IF(MI,J)=(N(1,J)         JCANILCAM, 1)=NL         IY=J=1         IF(J,EQ,NE)IX=1         IF(J,EQ,NE)IY=NE         JCANILCAM, 2)=HIL         JCANILCAM, 3)=HIL         ERMINE KOWS CONTA	NKS 1/100)*100.EQ.K) GU .1) GCTC 100 .J) .IN GCTC 100 .IX) .IN ING BINARY LINK	TU 20		
0005 0006 0008 0008 0011 0012 0013 0014 0015 0016 0017 0016 0017 0016 0017 0018	C C C C C C C C C C C C C C C C C C C	ICAM-1         RCH FOR BINARY LI         D0 10 1=1,NL         JCRUMI(CAM,1)=0         NE=P(1,NCCL)         D0 10 J=1,NE         IF(MI,J)-(MIL,J)-(M	NKS 1/1003+100.EQ.K) GU .1) GCTC 100 .J) .TY) .K) .TNING BINARY LINK	TU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014 0015 0016 0017 0016 0017 0016 0019 0020	C C C C C C C C C C C C C C C C	ICAM-1         RCH FOR BINARY LI         D0 10 1-1,NE         D0 10 1-1,NE         JCRUWIICAM,1)=         D0 10 J-1,NE         IF(MI,J)-(MI1,J)         CONTINUE         JCAMHICAM,1)-(MI1,J)         CAMHICAM,1)-(MI1,J)         CAMHICAM,1)-(MI1,J)         JCAMHICAM,1)-(MI1,J)         IF(MI,J)-(MI1,J)         JCAMHICAM,1)-MI1         IX-J-1         IF(J,JEQ,NE)IX=1         IF(J,FQ,NE)IX=1         JCAMHICAM,2)-MI1         JCAMHICAM,3)-MI1         ERMINE KOWS CONTA         JCROW(ICAM,1)=1	NKS 1/100]+100.EQ.K} GU .1) GCTC 100 .J) .IN .K) .IN BINARY LINK	TU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014 0015 0014 0015 0016 0017 0016 0017 0016 0019 0020	C C C C C C C C C C C C C C C C C C C	ICAM-1         RCH FOR BINARY LI         D0 10 1-1,NL         JCRUHICAM, 13=0         NE=P(1,NCCL)         D0 10 J-1,NE         IF(MI,J)-(MI1,J)         CONTINUE         IF(MI,J)-(MI1,J)         CANICAM, 13=M(1)         IX=J=1         IY=J-1         IF(J_EQ.NE)IX=1         JCAM(ICAM,2)=M(1)         JCAM(ICAM,2)=M(1)         JCAM(ICAM,2)=M(1)         PATHER KOWS CONTA         JCROW(ICAM,1)=1         JPT=1+1	NKS 1/100)#100.EQ.K) GO .1) GCTC 100 .J) .1Y) .IX) .INING BINARY LINK	TU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014 0015 0016 0017 0016 0017 0016 0019 0020	C C C C C C C C C C C C C C C	RCH FOR BINARY LI OC 100 K=1,NB DO 10 I=1,NL JCRUWIICAM,1]=NC IF(MI,NCCL) DO 10 J=1,NE IF(MI,J)-(M(1,J) CCNTINUE JCAM(1CAM,1]=M(1) IY=J=1 IF(J=EQ.NE)IX=NE IY=J=1 JCAM(1CAM,2)=M(1) JCAM(1CAM,2)=M1 JCAM(1CAM,3)=M1 BERMINE KOWS CONTA JCROW(1CAM,1)=1 IP1=I+1 DO 30 IA=IP1,NR	NKS 1/100)+100.EQ.K) GU .1) GCTC 100 .J) .IN .K) INING BINARY LINK	TU 20		
0005 0006 C007 0008 CC09 0011 0012 0013 0014 0015 0016 0017 C016 0017 C016 0019 C020	C C C C C C C C C C C C C C C C	ICAM-1         RCH FOR BINARY LI         D0 10 1-1,NL         JCRUHILGAM,13=0         NE=#(1,NCCL)         D0 10 J-1,NE         IF(MI,J)-(N(1,J)         CONTINUE         IF(MI,J)-(N(1,J)         CONTINUE         JCANGICAM,13=N(1)         Y=J-1         IF(J,EQ.NE)IX=1         IF(J,EQ.NE)IX=1         JCAMICAM,23=M(1)         JCAMICAM,23=M(1)         JCANICAM,23=M(1)         JCANICAM,23=M(1)         JCANICAM,23=M(1)         JCANICAM,21=1	NKS 1/100)*100.EQ.K) GU .1) GCTC 100 .J) .IV) .IX) .INING BINARY LINK	TU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014 0015 0016 0017 0016 0017 0016 0019 0020 0021 0022 0023 0024	C C C C C C C C C C C C C C C C	RCH FOR B(NAKY LI OC 100 K=1,NB DO 10 I=1,NL JCRUWI(CAM,1)=0 NE=P(1,NCCL) DO 10 J=1,NE IF(MI,J)-(M(1,J) CCNTINUE JCAH(1CAM,1)=M(1) IY=J-1 IY=J-1 IF(J_EQ_NE)IX=1 IF(J_EQ_NE)IX=1 IF(J_EQ_NE)IX=1 IF(J_EQ_NE)IX=1 IF(J_EQ_NE)IX=1 IF(J_EQ_NE)IX=1 JCAH(1CAM,3)=M1 JCROW(1CAM,1)=1 IP1=1+1 DO 30 IA=IP1,NR JCROW(1CAM,1)=0 NE=M(1A,NCD)	NKS 1/100)+100.EQ.K) GU .1) GCTC 100 .J) .IY) .IX) NINING BINARY LINK	ΤU 20		
0005 0005 0008 0010 0011 0012 0013 0014 0015 0016 0017 0016 0017 0016 0019 0020	C SEAN C SEAN C 10 20 C DETI	ICAM-1         RCH FOR BINARY LI         D0 10 1=1,NL         JCRUHILCAM, 1)=0         NE=P(1,NCCL)         D0 10 J=1,NE         JCRUHILCAM, 1)=(NL1,J)         D0 10 J=1,NE         JCRUHILCAM, 1)=(NL1,J)         CONTINUE         IF(MI,J)=(NL1,J)         JCANILCAM, 1)=(NL1,J)         IX=J=1         IY=J=1         IF(J,EQ.NE)IX=1         JCANILCAM, 2)=HI         JCANILCAM, 2)=HI         JCANILCAM, 2)=HI         JCANILCAM, 3)=HI         JCANILCAM, 3)=HI         JCROWILCAM, 1)=1         IP1=1+1         D0 30 LA=IP1,NR         JCROWILCAM, IA, NCOL)         YEINE, EN ON GOOTA	NKS 1/100)+100.EQ.K) GU .1) GCTC 100 .1) .17	ŦU 20		
0005 0006 0007 0008 0010 0011 0012 0013 0014 0015 0016 0017 0016 0017 0016 0019 0020 0021 0022 0023 0024 0025 0026	C C C C C C C C C C C C C C C C	RCH FOR BINARY LI OC 100 K=1,NB DO 10 I=1,NL JCRUWIICAM,1]=0 IF(NI,J)=(M(1,J) CCNTINUE IF(NI,J)=(M(1,J) CCNTINUE IF(NI,J)/100.NE JCAM(1CAM,1]=M(1) IF(J_EQ.1)IY=NE JCAM(1CAM,2)=M(1) JCAM(1CAM,2)=M(1) JCAM(1CAM,3)=M(1) JCAM(1CAM,3)=M(1) JCAM(1CAM,1)=1 JCAM(1CAM,1)=1 JCAM(1CAM,1)=1 JCAM(1CAM,1)=1 IP(=1+1) DO 30 1A=1P1,NR JCROW(1CAM,1A)=0 NE=M(1A,NCU) IF(NE,EQ.C) GGTO	NKS 1/100)+100.EQ.K) GU .1) GCTC 100 .J) .IV) .IX) NINING BINARY LINK	TU 20		
0005 0006 C007 0008 CCC9 0011 0012 0013 0014 0015 0016 0017 C016 0017 C016 0019 C020 0021 0022 0023 0024 0025 0026 0027	C C C C C C C C C C C C C C	RCH FOR BINARY LI DC 100 K=1,NB DO 10 I=1,NL JCRUWIICAM,I)=0 NE=P(1,NCCL) DO 10 J=1,NE IF(MI,J)=(N(1,J) CCNTINUE IF(MI,J)=(N(1,J) CCNTINUE IF(MI,J)=(N(1,J) IF(J)=CQ.NE)IX=1 IF(J)=CQ.NE)IX=1 IF(J)=CQ.NE)IX=1 IF(J)=CQ.NE)IX=1 JCAM(ICAM,2)=HI JCAM(ICAM,2)=HI JCAM(ICAM,2)=HI JCAM(ICAM,2)=HI JCAM(ICAM,2)=HI IF(J)=I CONTICAM,1)=1 IF(J)=CQ.NE JCROW(ICAM,1)=1 IF(NE,EQ.C) GGTO DC 31, IB=1,NE	NKS 1/100)*100.EQ.K) GU .1) GCTC 100 .1) .1) .1) .1) .1) .1) .1) .1)	TU 20		
0005 0006 C007 0008 CC09 0011 0012 0013 0014 0015 0016 0017 C016 0019 C020 0021 0022 0023 0024 0025 0026 0027 0028	C C C SEAI C 20 C C DETT	RCH FOR B(NAKY LI OC 100 K=1,NB DO 10 I=1,NL JCRUWI(CAM,1)=0 NE=P(1,NCCL) DO 10 J=1,NE IF(M(1,J)=(M(1,J) CCNTINUE IF(M(1,J)/100.NE JCAM(1CAM,1)=M(1) IF(J=C,NE)IX=1 IF(J=C,NE)IX=1 IF(J=C,NE)IX=1 IF(J=C,NE)IX=1 IF(J=C,NE)IX=1 JCAM(1CAM,2)=M(1) JCAM(1CAM,2)=M(1) JCAM(1CAM,1)=1 JCAM(1CAM,1)=1 JCAM(1CAM,1)=1 JCAM(1CAM,1)=1 IP(=1:+1 DO 30 1A=1P1,NR JCROW(1CAM,1A)=0 NE=M(1A,NCUL) IF(M(1A,1B)=EQ.)	NKS 1/100)+100.EQ.K) GO .1) GCTC 100 .1) .1Y	TU 20		
0005 0006 C007 0008 CCC9 0011 0012 0013 0014 0015 0016 0017 C016 0017 C016 0019 C020 0021 0022 0023 0024 0025 0026 0027 0028 0029	C SEAN C SEAN C 10 20 C DETT C DETT	ACH FOR BINARY LI         D0 10 1=1,NL         JCRUWIICAM,1)=0         NE=P(1,NCCL)         D0 10 J=1,NL         JCRUWIICAM,1)=(N11,J)         D0 10 J=1,NE         IF(M1,J)-(N(1,J)         CCNTINUE         JCAH(1CAM,1)=M(1)         IF(M1,J)-(N(1,J)         CCNTINUE         JCAH(1CAM,1)=M(1)         IX=J=1         IY=J=1         IF(J,EQ.NE)IX=1         JCAM(1CAM,2)=H1         JCAM(1CAM,2)=H1         JCAM(1CAM,3)=H1         DO 30 1A=1P1,NR         JCROW(1CAM,1A)=1         JP1=I+1         D0 30 1A=1P1,NR         JCROW(1CAM,1A,A)         JCROW(1CAM,1A)=0,CL)         JF1NE.EQ.JCGUTO         DC 31 IB=1,NE         IF(M.E.Q.C)         JCMI 1CAM,1B)=LG.J         CONTINUE	NKS 1/1003+100.EQ.K) GU .1) GCTC 100 .1) .1) .1) .1) .1) .1) .1) .1)	TU 20 2		
0005 0006 C007 0008 CC09 0011 0012 0013 0014 0015 0016 0017 C016 0019 C020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0029 0029	C C C SEAI C 20 C C DETI C 31	RCH FOR BINARY LI OC 100 K=1,NB DO 10 I=1,NL JCRUWIICAM,1]=O IG JOINE IF(MI,J)=(MI1,J) CCNTINUE IF(MI,J)=(MI1,J) CCNTINUE IF(MI,J)=(MI1,J) CCNTINUE IF(MI,J)=(MI1,J) IF(J,EQ,NE)]X=1 IF(J,EQ,NE)]X=1 IF(J,EQ,NE)]X=1 IF(J,EQ,NE) IF(J,EQ,NE)]X=1 IF(J,EQ,NE) IF(J,EQ,NE) IF(J,EQ,NE) IF(J,EQ,NE) IF(J,EQ,NE) IF(J,EQ,NE) IF(J,EQ,NE) IF(MIA,NCU) IF(MIA,1B),EQ,J) CONTINUE GGTC 30	NKS 1/100)*100.EQ.K) GO .1) GCTC 100 .1) .17	TU 20 2		
0005 0006 C007 0008 CCC9 0011 0012 0013 0014 0015 0016 0017 C016 0017 C016 0017 C020 0021 0022 0023 0024 0023 0024 0025 0026 0027 0028 0029 0031	C SEAN C SEAN C 10 20 C DETT C 31	RCH FOR BINARY LI DC 100 K=1,NB DO 10 I=1,NL JCRUWIICAM,1)=0 NE=P(1,NCCL) DO 10 J=1,NE IF(MI,J)-(M(1,J) CCNTINUE JCAM(1CAM,1)=M(1) IY=J=1 IF(J=CQ.NE)IY=NE IF(J=CQ.NE)IY=NE IF(J=CQ.NE)IY=NE JCAM(1CAM,2)=M(1) JCAM(1CAM,2)=M(1) JCAM(1CAM,2)=M(1) JCAM(1CAM,1)=1 IP(J=1+1) DO 30 1A=IP1,NR JCROW(1CAM,1A)=0 NE=M(1A,NCOL) IF(NE=CQ.O) GDTO DC 31 IB=1,NE IF(M[1A,IB)=CG.J GGTC 30 JCROW(1CAM,1A)=0	NKS 1/1003+100.EQ.K} GU .1) GCTC 100 .J) .(X) .INING BINARY LINK 	TU 20 2		
0005 0006 C007 0008 CC09 0011 0012 0013 0014 0015 0016 0017 C016 0019 C020 0021 0022 0023 0024 0025 0026 0027 0028 0029 C030 C031 0032	C SEAN C SEAN 10 20 C DETI C DETI 31 32	RCH FOR BINARY LI DC 100 K=1,NB DO 10 I=1,NL JCRUHI(TAM,I)=0 NE=V(1,NCCL) DO 10 J=1,NE IF(MI,J)=(N(1,J) CCNTINUE IF(MI,J)=(N(1,J) CCNTINUE IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 IF(J,EQ,NE)IX=1 JCAN(ICAM,1)=1 IPI=I+1 DO 30 1A=IPI,NR JCROW(ICAM,I)=1 IF(MIA,IB)=EQ,J CONTINUE GGTC 30 JCROW(ICAM,A)=1 CCNTINUE	NKS 1/100)*100.EQ.K) GU .1) GCTC 100 .1) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y .1Y) .1Y .1Y) .1Y .1Y .1Y .1Y .1Y .1Y .1Y .1Y	TU 20 2		
0005 0006 C007 0008 CC09 0011 0012 0013 0014 0015 0016 0017 C016 0017 C016 0019 C020 0021 0022 0023 0024 0023 0024 0025 0026 0027 0026 0027 0026 0027 0026	C SEAN C SEAN 10 20 C DETT C DETT 31 32 30	RCH FOR BINARY LI DC 100 K=1,NB DO 10 I=1,NL JCRUWI(CAM,1)=0 NE=P(1,NCCL) DO 10 J=1,NE IF(MI,J)-(M(1,J) CCNTINUE JCAM(1CAM,1)=M(1) IY=J=1 IF(J=EQ.NE)IY=NE IF(J=EQ.NE)IY=NE IF(J=EQ.NE)IY=NE IF(J=EQ.I)IY=NE JCAM(1CAM,1)=1 IPI=I+1 DO 30 IA=IPI,NR JCROW(1CAM,1A)=0 NE=M(IA,NC) IF(MIA,1B)=EQ.J JCROW(1CAM,1A)=I IF(MIA,1B)=EQ.J JCROW(1CAM,1A)=I ICONTINUE GGTC 30 JCROW(1CAM,1A)=I CCNTINUE	NKS 1/1003+100.EQ.K) GU .1) GCTC 100 .3) .11) .12) .13) .11) .13) .13) .14) .15) .15) .100 .13) .14) .15	TU 20 2		
00005 00005 00008 00011 0012 0013 0014 0015 0016 0017 0016 0017 0016 0017 0020 0021 0022 0023 0024 0025 0026 0027 0026 0027 0028 0026 0027 0028 0029 0030 0031 0032 0033	C SEAN C SEAN C DETI C DETI 31 32 30	ICAM-1         RCH FOR BINARY LI         D0 10 1-1,NL         JCRUHILGAM,13-0         NE=P(1,NCCL)         D0 10 1-1,NE         JCRUHILGAM,13-0         IF(MI,1,3)-(M(1,3	NKS )/100)+100.EQ.K) GU .1) GCTC 100 .1) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y) .1Y .1Y) .1Y .1Y) .1Y .1Y) .1Y .1Y) .1Y .1Y .1Y .1Y .1Y .1Y .1Y .1Y	TU 20 2		· · · · · · · · · · · · · · · · · · ·
0005 0006 C007 0008 CC09 0011 0012 0013 0014 0015 0016 0017 C016 0017 C016 0019 C020 0021 0022 0023 0024 0023 0024 0025 0026 0027 0026 0027 0028 0026 0029 C030 C031 0032 0034	C SEAN C SEAN 10 20 C DETT C DETT 31 32 30 100	RCH FOR B(NAKY LI DC 100 K=1,NB DO 10 I=1,NL JCRUWI(CAM,1)=0 NE=P(1,NCCL) DO 10 J=1,NE IF(MI,J)-(M(1,J) CCNTINUE JCAM(1CAM,1)=M(1) IY=J=1 IF(J=EQ.NE)IY=NE JCAM(1CAM,1)=1 IF(J=EQ.NE)IY=NE JCAM(1CAM,1)=1 IP(J=EQ.I)IY=NE JCAM(1CAM,1)=1 IP(I=I+1 DO 30 IA=IPI,NR JCROW(1CAM,1)=1 IF(MIA,1B)=EQ.J) CONTINUE GGTC 30 JCROW(1CAM,1A)=1 CCNTINUE ICAM=ICAM+1 CCNTINUE	NKS 1/100)+100.EQ.K) GU .1) GCTC 100 .1) .1Y) .1Y) .1X) INING BINARY LINK 	TU 20 2		
00005 00005 00008 0001 0011 0012 0013 0014 0015 0016 0017 0016 0017 0016 0017 0016 0017 0020 0021 0022 0023 0024 0025 0026 0027 0026 0027 0026 0027 0026 0027 0026 0027 0026 0027 0026 0023 0024 0025	C SEAN C SEAN C DET C DET C 31 32 30 100	ICAM-1         RCH FOR BINARY LI         D0 10 1-1,NE         D0 10 1-1,NE         JCRUHILCAM, 1)=ON         NE=P(1,NCCL)         D0 10 1-1,NE         JCRUHILCAM, 1)=(NL1,J)         JCATINUE         IF(MI,J)=(NL1,J)         JCATINUE         JCATINUE         JCATINUE         JCANIICAM, 1)=(NL1,J)         JCANIICAM, 1)=(NL1,J)         JCANIICAM, 1)=(NL2,EQ,NL2)         JF(J=EQ,NE)IX=1         IF(J=EQ,NE)IX=1         JCANIICAM, 2)=(NL1,J)         JCANIICAM, 2)=(NL1,J)         JCANIICAM, 2)=(NL1,J)         JCANIICAM, 2)=(NL1,J)         JCROWIICAM, 1]=1         JPI=1+1         D0 30 1A=1P1,NR         JCROWIICAM, 1]=N,I         JCROWIICAM, 1]=1,NR         JCROWIICAM, 1]=1         IF(NE_EQ_C)         GGTO 30         JCROWIICAM, 1]=1         ICAM=ICAM=1         CONTINUE <t< td=""><td>NKS )/100)+100.EQ.K} GU .1) GCTC 100 .1) .17</td><td>TU 20 2</td><td></td><td>·</td></t<>	NKS )/100)+100.EQ.K} GU .1) GCTC 100 .1) .17	TU 20 2		·
0005 0006 C007 0008 CC09 0011 0012 0013 0014 0015 0016 0017 C016 0019 C020 0021 0022 0023 0024 0023 0024 0025 0026 0027 0026 0027 0026 0027 0026 0027 0026 0023 0024 0025 0026 0025 0026 0027 0026 0027 0026 0023 0026 0027 0026 0027 0026 0027 0026 0027 0011 0012 0013 0014 0015 0016 0011 0012 0013 0014 0015 0016 0017 0016 0020 0021 0022 0023 0024 0023 0024 0023 0024 0023 0024 0025 0026 0023 0026 0023 0024 0023 0024 0025 0026 0023 0024 0023 0024 0025 0026 0023 0024 0025 0026 0023 0024 0025 0026 0025 0026 0023 0024 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 0026 0025 00026 0025 00026 00026 00025 00026 00025 00026 00026 00025 0000000000	C SEAI C SEAI 10 20 C DETI C DETI C 31 32 30 100	RCH FOR B(NAKY LI OC 100 K=1,NB DO 10 I=1,NL JCRUWI(CAM,1)=0 NE=P(1,NCCL) DO 10 J=1,NE IF(MI,J)-(M(1,J) CCNTINUE JCAH(1CAM,1)=M(1) IY=J=1 IY=J=1 IF(J=EQ.NE)IY=NE JCAH(1CAM,1)=1 IF(J=CQ.I)IY=NE JCAM(1CAM,1)=1 IP1=I+1 DO 30 IA=IP1,NR JCROW(1CAM,1)=1 IP(NE=EQ.C) GGT 30 JCROW(1CAM,1A)=0 IF(MIA,1B)=EQ.J CONTINUE ICAM=ICAM=1 CCM=ICAM=1 RETURN	NKS 1/100)+100.EQ.K) GU .1) GCTC 100 .1) .1Y] .1Y	TU 20 2		
00005 00006 C007 0008 CCC9 0011 0012 0013 0014 0015 0016 0017 C016 0017 C016 0017 C016 0019 C020 0023 0024 0023 0024 0025 0026 0027 0026 0027 0026 0027 0030 C031 0032 0031 0032 0034 0035 0036 0037	C SEAN C SEAN C DET C DET 31 32 30 100	RCH FOR BINARY LI DC 100 K=1,NB DO 10 I=1,NL JCRUWI(ICAM,1)=0 NE=P(1,NCCL) DO 10 J=1,NE IF(MI,J)=(N(1,J) CCNTINUE IF(MI,J)=(N(1,J) CCNTINUE IF(J,EQ.NE)IX=1 IF(J,EQ.NE)IX=1 IF(J,EQ.NE)IX=1 IF(J,EQ.NE)IX=1 IF(J,EQ.NE)IX=1 JCAM(ICAM,2)=HI JCAM(ICAM,2)=HI JCAM(ICAM,2)=HI JCAM(ICAM,2)=HI JCAM(ICAM,1)=1 IP(I=I+1 DO 30 IA=IP1,NR JCROW(ICAM,IA)=1 IF(NE,EQ.C) GOTO DC 31 IB=1,NE IF(NE,EQ.C) GOTO CC 31 IB=1,NE IF(NE,EQ.C) GOTO CC 31 IB=1,NE IF(NE,EQ.C) GOTO CC 31 IB=1,NE IF(NE,EQ.C) GOTO CC 31 IB=1,NE IF(NE,EQ.C) GOTO CCNTINUE ICAM=ICAM=1 RETURN RETURN	NKS 1/1003+100.EQ.K) GU .1) GCTC 100 .1) .17	TU 20 2		·





## TABLE LII

## SUBROUTINE CAM3

FORTRAN IN	G LEV	/EL	L, MOD	4	CAM3	DAT	TE = 70193		14/26/56		PAGE 0001	
0001			SUBROU	TINE	CANS							
0002			IMPLIC	1 11	NTEGER#2(I-K)							
0003			COMMON	ALL	/#(15,15).NB.NL.NR.N	COL .NK .NCH	NELEN.NSTO	RE				
0004			CONMGN/DBALL/JSTORE(100,6), JTEMP(100,6)									
0005			COMMON	/DBL	MX/MZ(15,15),MZ1(15,	15), MZ 2( 15)	,15),MZ3(15	151,M	24(15,15)			
			++HZ5(1	5.15	1,MX(50,15,15)							
0006			COMMON	/CAL	/JCAM(8,3), JCROW(8,1)	5)						
0007			COMMON.	/CHN	I/ICAM,KSTORE,KOUNT							
6008			COMMON	/LIN	/LINE, IPAGE							
0009			ICAN=1									
0010			KOUNT=	1								
0011			KSTORE	=1						,		
0012			1F(LIN	E•GT	.47) CALL PAGE							
0013			WRITE	6,1)								
0014		1	FORMAT	<u>.</u>	OIDENTIFICATION OF C	HAINS WITH	THREE CAM	PAIRS"	•/•1ו			
		4	45(***	1./1								
0015			LINE	INC								
0010			DO TOO		INSTURE							
0017			JICAPI	1	- STORELLSZ							
0010		100	JIEARI		-JSTURE(I)IJ							
0019			16/17	14-1 60 1	LINSTORE							
0020	5 C		161 176	69°1	7.1).NG. (TEMO/17-1.1							
0022			COTO A		21111112-111							
0022		a	I = ITEN									
0024			LieJCA	M(1.	11							
0025			CALL	AMAD	DIL.NR.NCOL.M.HZI)							
0026		8	J=JTEM	PIIZ	-21							
0027		-	JI=JCA	H(J,	11							
0028			CALL C	AHAD	ULJ.NR.NCUL.NZ1.NZ2)							
0029			JP1=J+	1								
0030			IF(JP1	.GT.	NCH) GOTO 10							
0031			DQ 30	K=JP	1.NCH	1						
0032			K 2= JCA	мεк,	23							
0033			K3=JCA	HLK,	3)							
0034			1F([]+	EQ.K	(2.OR.11.EQ.K3) GOTO :	30						
0035			IF(J1.	EC.K	2.0R.J1.EQ.K3) GOTO	30						
0036			JSTCRE	(ICA	M,13=1						•	
0037			JSTORE	LICA	M,23=J							
0038			JUNE	LISA	M-33=K							
0039			KUUNI*	KUUP								
0040			CALL C	A M A C	DIADI CALL CHAINI							
0041			WE LL L	ARAL	JULK JAK JACOL JAZZ JAZJ							
0042			1900-1									
0042			BO 20	1 4=1	- NR							
0044	·		TELICE	<u>awr</u>	.IA) .NE. LA.AND. ICROW	LISTAD NE-1	LA.AND.JCRC	16 IK. FA	JANEALAL			
			GOTO	20								
0045			00 40	18=1	.NCOL							
0046		40	HXI ICA	H, LA	QW, [B)=MZ([A, [B]							
0047			IRCW=1	ROW	• <b>1</b>							
6048		20	CONTIN	UE								
0049			HX{1CA	H, 1,	NCOL-1)=1ROW							
0050			ICAN=I	CAH	1							
0051		30	CONTIN	UE								
0052		10	CONTIN	UE								
0053			ICAN=1	CAM-	-1							
0054			IFIICA	M.GE	E-KSTURE) CALL CHAINI							
FORTRAN IN	GLEV	/EL	1, MOĐ	4	CAN3	GAT	TE = 70193		14/26/56		PAGE GOOL	
0055			NCH-IC									
0056			NSTORE	=NCH	1 · · · · · · · · · · · · · · · · · · ·							
0057			RETURN									
C056			END									

NOTE: Subroutine CAMADD appears in Table LVI.

,



## Figure 29. Flow Chart for Subroutine CAM3

### APPENDIX J

# SUBROUTINES USED IN DETERMINING THE TEN-LINK CHAINS WITH PRISM PAIRS

The program for prism pairs follows the same form as all previous programs. Subroutines JOINT (which locates all joints) and PRISM2 (which adds 2 prism pairs) are shown in Tables LIII and LIV respectively. Their flow charts appear in Figures 30 and 31.
# TABLE LIII

# SUBROUTINE JOINT

FORTRA	N IV G LEVEL	1, MOD 4	JO INT	DATE = 70195	02/53/33	PAGE 0001
0001		SUBROUTINE JO	INT			
0002		IMPLICIT INTE	GER+2(1-N)			
0003		COMMON/ALL/NI	15.15), NB, NL, NR, NCC	L,NK,NCH,NELEM,NSTORE		
0004		COMMON/JO1/JL	DC(13,2),JROW(13,15	; ;		
0005		NUH=13				
0007		1 2#2				
COCE		JNUM=1				
0009		00 1 1=1.13				
0010		DC 1 J=1,15				
0011	1	JROW(1,J)=0				
	C	· · · · · · · · · · · · · · · · · · ·				
	C SEA C	RCH FOR EITHER	LINK LI OR LINK LA	2•		
0012	2	1SUM=0				
0013		0C 10 I=1,NR				
0014		NE=H[[,NCOL]				
0015		1F(NE.EQ.0) G	UTO 10			
0010		10 20 J=1+NE	1.11/1001#100 50 0	1 6010 30		
0018		1F(H(1,1)+1H)	1.11/1001+100.EQ.L1			
0019	20	CONTINUE	-,-,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			
0020		GOTO LO				
0021	30	1F(M(1,J+1)-()	H{I,J+1}/100}+100_E	Q.L2) GOTO 50		
0022		IF(J.NE.1) GO	TO 10			
0023		1F(M(1,NE)-(M	{ I,NE}/100}+100.EQ.	L2) GOTO 60		
0024		GOTO 10				
0025	40	1571 NC 11 CO	M(1,J+11/100)+100.	GALLI GUIU /0		
0023		1F(J+NE+1) 00	(1.NE)/1001#100.E0.			
0028		GOTO 10				
	C,			3		
	C LIN	KS LI AND L2 A	RE ADJACENT IN ROW	1.		
0029	50	JLCC(JNUM,1)=	HELEJJ			
0030		JLOC(JNUH,2)=	MEI,J+1)			
0031		GCTO 90				
CC 32	60	JUGG(JNOP+1)=	7(1)J] W/T NEL			
0034		- JEUEIJNUM;2/=0 - CCTD 90	41 1 1 AUE 1			
0035	70	JLUC(JNUH.1)=	H{1.J+13			
0036		JLUC (JNUM, 2)=	M(1.J)			
CC37		GOTO 90				
86 00	.80	JLOC(JNUM,1)=	M(I,NE)			
0039		JLUC (JNUM, 2)=	M(1+1)			
004¢	90	JROWIJNUM:II=	l			
0041	10	15UM=1 CONTINUE				
0042	10	LEFTSIN ED AL	6010 100			
0044		JNUM#JNUM#1	0010 100			
0045		IF(JNUM.GT.13	) RETURN			
0046	100	L2=L2+1				
0047		IF(L2.GT.NB)	GOTO 110			
CC48		GCTO 2				
0049	110	Ll=Ll+l				
0050		L2#L1+1				
0051		GOTO 2	REIUKN			
0052						

a and a second





#### TABLE LIV

### SUBROUTINE PRISM2

FOR TRAN	IV G	ĻĒ	VEL	1. MC	04	PRIS	M2	UATE =	70195	02/53/3	13	PAGE	0001
0001 0002 0004 0005 0006 0007 0008 0009 0010 0011 0012 0013				SUBRON IMPLIA COMMON COMMON COMMON COMMON COMMON IPRIS KOUNTA KSTORI IF(LI)	UTINE CIT IN N/ALL/ N/DBALI N/DBALI N/DBALI N/DBALI N/DBALI N/DBALI N/DBALI N/DBALI N/DBALI N/DBALI N/DBALI N/DBALI N/DBALI	PRISM2 FEGER+2(I-N) 4(15,15),NB,7 /JSTURE(100, M/M2(15,15),F PAIR(13,2),I PRISM,KSTORE LINE,IPAGE	NL, NR, NC DL, N 6), JTEMP(10 721(15, 15), M RCW(13, 15) ; KOUNT	K,NCH,NELI 0,6) X1100,15,	EM, NSTORE 15)		·		
0014			1	FORMA1	r:/,•0 •3,/)	IDENTIFICATIO	DN OF CHAINS	WITH TWO	PRISH PAI	IR\$*•/•1X•			
0015 0016 0C17 0018 0019 0020 0021 0022 0022			100 9	LINE=1 DO LOO JTEMP4 DU LO I=JTE1 CALL 1 TP1=14 IF(IP) DO 30	LINE+5 D I=1, [1,1]= IZ=1, PP(IZ, LOCATE +1 I.GT.N J=IP1	NSTORE JSTCHE(1,1) NSTCRE 1) (1,NR,NCOL,M, CH) GOTO 10 NCH	MZ1 3				•		
0024				JSTOR	ECIPRI ECIPRI	SM,1}=I 5M,2}=J							
0028				IF(KO)	UNT.GT	FI •53 CALL CHAI (J•NR•NCOL≠N)	IN 1 21 - MZ 1						
0029 003C 0031		C	FIL	TROW=1 DO 20 IF(LRO	1 	NR N - E 9 - 0 - AND - L	.RCW[J,[A].E	Q.0) GOTO	20				
0032			40	DU 40 MX(IP) IRNU=	18=1, R1SM,[: 1909+1	NCUL {OK,IB}=NZ(I/	,1B)						
0035			20	CONTI	NUE RESM.1	NCOL-1]=IRO		,					
0037 CC38			30	IPRISE CONTI	N=IPRI: NUE	5M+1							
0040			10	IPRIS	NDE M≈IPRI RISM∎GI	SM+1 E.KSTORE) CAI	L CHAINL						
0042 0043				NCH= II RE TURI	PRISM N								
- 0044				END					:			1 A.	
FORTRAN	IV G	LE	VEL	1 # - MO(	4	LOCA	AT E	DATE =	70195	02/53/3	3	PAGE	0001
0001 0002 00C3 0004 CCC5 0006				SUBRUL IMPLIC CUMMON DIMENS DG 10 IF(JRC	JTINE   CIT IN N/JOI/ SION M° I≠1,N DW(JN,	DCATE(JN,NR) FEGER#2(IN) JLUC(13,2),JF (I15,20),MZ(J R ().EQ,0) GOTO	NCOL:MY,M2) ROW(13,15) L5,20) ) 18						
		C C C	\$E A	RCH F01	R LENK	S INCLOENT TO	NL THIDL						
0007 0CC8 0009 0010 0011 0012 0C13		L	20 40	NE=MY1 DO 20 IF(MY1 IF(MY1 CONTIN IF(MY1 GBTD 1	(1,NCO) J=1,N (1,J), (1,J), NUE (1,J+1 NUE (1,J+1	.) EQ.JLOC(JN,1) EQ.JLOC(JN,2)	3 GOTO 40 3 GOTO 50 233 GOTO 60						
0014 0015			50	LF ( MY) Goto T	1 <b>1,J+1</b> 70	I • E Q• JLOC ( JN)	1)) GCTO 60						
		c c	IN S	ERT PRI	ISM PA	IR IN PROPER	LUCATION						
CC16 0017 0018 CC19 0020 0021 0022 0023 0023 0024 0025			70 60 61	K\$*NE J=0 GOTC 6 KS=J MZ{1,J NE=NE J=J+1 D0 62 MZ{1,} J=J+1	51 L]=101 +1 K=2,N (}=PY{								
0026 0027 0028 0029 0030 0031 0032 0033 0034			62 03 64	IF(J.) IF(J.) CONTIN K=K+1 DO 64 MZ(1,) MZ(1,) GOTO 19 MZ(1,)	EQ.KS+ EQ.NE} NUE IK=K, [K]=O NCOL}= 10 J=1,N	I) GOTO 63 J≠1 NCOL NE COL							
0036 0037 0038			10	CONTIN RETURN END	NUE N	.,.,							





#### APPENDIX K

# SUBROUTINES USED IN DETERMINING THE TEN-LINK CHAINS WITH DOUBLE JOINTS

This program follows the same form as all previous programs. Subroutine DBLOC (Table LV) locates all possible double joints. The flow chart for Subroutine DBLOC appears in Figure 32. Subroutine DBL2, which forms modified MPM's for chains with two double joints, is shown in Table LVI and its flow chart in Figure 33. In addition, Subroutines LNKCNT and MOBCK, which check for mobility of the chain by applying rules 1 and 2 of Chapter X, are shown in Table LVII. Their flow charts appear in Figure 34.

# SUBROUTINE DBLOC

	FORTRAN	IV G	L	EVEL	1,	NCD	4	DBLCC	٥	ATE -	70199	08/35/08	PAGE	0001
	0001				SUB	ROU	TINE COLO	DC						
	0003				CON	4UN	/ALL/Mil	5,15),N8,NL,NR,I	COL .NK .NC	H,NEL	.EM, NSTOR	E		
	0004				COM	40N	JOBL/JLO	C(20,3),JDROW(20	0,15),JROW	( 20, 1	.5)			
			c	SEA	RCH	FOR	BASE LI	NKS						
	0006				NCH-	-1	I Aw L.NR							
	0008				00	ii.	1=1.NL							
,	0010				NE= 00	PCI 11	,NCGLJ J=1,NE							
	0011				151	201	11HI-11	.J)/100)+10C.NE	LNI GOTO	11				
	0013				GOT	5 1	2	C1.37 GC16 10						
	CC14		c	11	CCN K IN	11N 15	UE A HASE I	LINK. FIND ALL	LENKS JOI	NED T		1.NK		
	0015		Č	12	1BA	SE=	H(1,J)							
	0016 0017			13	LST	L3 Jre	15=1+5 (15)=0							
	0018				15=									
	0020			20	1 X=.	J−1 J+1								
	CC21				IF	J.€	Q.1) 1X=0	NE 1						
	0023				00	21	IA=1,IS	•						
	CO24 0025			21	1Ft CON	4 C E F E N	+IX}.EQ.  UE	LSTCRE(IAJ) GCT	22					
	0028			_	LST	CRE	(1S)=M(1)	,IX)						
	0028				IFI	ls.	GT.IBASE	/1003 6010 100						
	0029			22	00	23	18=1+15	STORF(18)) 6010	30					
	0031			23	CON	F I N	UE							
	0032				15=	DAE 15+	[[S]=M([ 1	1143						
	0034			•••	IFL	is.	GT.IBASE	/100) GGTG 100	·					
	0035			30	00	1 + 1 36	I=11,NR							
	0037				NE=	H(1	INCCLI	10 36						
	0039				EC.	35	J=1,NE						•	
	0040			35	LFE	46 E F I N	.↓].€Q.[  UE	BASE) GOTO 20						
	0042		~	36	CON	II.	UE		-	ALC: N	NETCOMIN	CALL CONSTRATIONS		÷.,
			č	OF	THES	LS E L	INKS TAK	EN THO AT A TIM	E 8228 FUU E.	NU -	DELENATA	C ALL CUMBINATIONS		
	CC43			100	IS=	I S-	1					*		
	0045				0C	110	1A=1+15	HL						
	0046				1AP DD	L=1 L L O	A+1 18=1AP1	. 15						
			ç	INS	URE	<b>FHA</b>	T AT LEA	ST FOUR LINKS W	ILL REMAIN	INE	ACH LOOP	CONTAINING THE		
	0048		L	QUU	CC	101	JA=1,NR							
	0049				IFC ISU	41 J 4=0	A,NCUL).(	GT.4) GCTO 101						
	0051				00	102	JB=1,4							
	.çuşz				11-		* ( JO )							
	CRTRAN	1V G	L	EVEL	1.	•00	4	DELOC	D	ATE -	70199	08/35/08	PAGE	0002
	0053			102	LEC	12.	EQ. LBASE	.UR. IZ. EQ. L STCR	E(IA).OR.I	Z.EQ.	LSTORELI	B))15UH= ISUH+1		
	0054			101	IFI	SU	H.E4.3) (	GUTO 110						
	0056			101	JLD	( N	ос Сн,1}=18/	ASE						
	0057				JLO(	24 N 24 N	CH#2]=LS1 CH#3]=LS1	TORE([A] Tore([8]						
			c	LOCA	ATE	LL	RONS IN	VOLVING JOINT NO	Сн.					
	0059				0C	132	1=1,KR							
	0061				JORI	) w (	JN+[]=0						* •	
	0063				NE=	qī	NCGL)						•	
	0064				IF() DC	ν£. 130	EQ.0) GO1 J=1.NE	TO 132	÷.					
	0066				IFU	11	, J ) . NE . JI	LOC(JN,1)} GOTO	130					
	0067				1 -	j+1					¢-			
	0069				IFC	).E	Q.[] IX=7 Q.NE] IX+	NE • 1			•			
	0071				IFU	<u>ii</u>	.1X1.EQ	JLOC ( JN. 2) - AND-	111,1Y).EQ	. JL00	(JN.3))	GCTC 133		
	0072				164	***	,[X].EQ	JLCC ( JN , 2 ) - CR + M	(1,1X).EQ.	JLOCI	(JN:3)) G	OTO 131		
	0074				IFU	11	111.EQ.	JLOC(JN,2).OR.M	[1,1Y].EQ:	JLOC	JN.3)1 G	GTO 131		
	0076			133	JRO		N,1)=1				•	· · ·		
	0077			131	JOR	) W E [ ] N	JN.IJ=I Ué							
	0079			132	CON	TIN	UE							
	0080				LE LE	- M 108	LEG.0) G	TO 110						
	0082			110	NCH	NC	H+1							
	0084			10	CON	ri N	UE							
	0085 0085				RET	INC JRN	# <b>- 1</b>							
	0087				END									



Figure 32. Flow Chart for Subroutine DBLOC

### TABLE LVI

# SUBROUTINE DBL2

FORTRAN IV	G LEVEL	1, MCD	•	C8L2	DATE	- 70199	08/35/08	PAGE 0001
0001		SUBKOUT	INE DBL 2					· · · ·
COC2		IMPLICI	T INTEGER#2	2(1-N)				
0003		COMMONZ	ALL/H(15,19	SI NB NL NR NC	OL , NK , NCH , KE	LENINSTORE		
0004		COMMUN/	OBALL/JSTO	REI 100,61, JTEM	P(100,6)			
CCCS		CORMON/	DBLFX/FZ(19	5,15),NZ1(15,1	5),HZ2(15,15	),#Z3(15,15	),MZ4(15,15)	
0006		COMMON/	DBL/JLCC(20	3,3), JCRCw (20,	15), JROW (20,	15)		
0007		COMMUNZ	CHN/IDBL,KS	STOKE KOUNT				
0008		COMMON/	D8L6A/11(3)	},J1(3),K1(3),	L1(3),M1(3),	N1(3)		
0009		COMMON/	LIN/LINE,II	PAGE			· ·	
0010		1CBL=1						
0011		KCUNT#1						
0012		KSTORE=	1					
0013		IFIL INE	GT.473 CAL	LL PAGE				
0014		<b>WRITE(6</b>	,1)					
0015	1	FORMATE	, OLDENTI	FICATION OF CH	AENS WITH TH	O DOUBLE JO	1NTS',/,1X,	
		* 476***	1./)					
CC16		LINE=LI	hE+5					
0017		00 100	I=1,NSTORE					
0018	100	JTEMPLI	.1)=JSTORE(	(1.1)				
0019		DC 10 L	Z=1,NSTCRE					
0020	9	[=JTEMP	(12,1)					
0021		CALL CB	LADD(I,NA,M	COL:M.MZ1)				•
0022		101=1+1						
0023		IF(191.	GT.NCHI GO1	TO 10				•
0024		DO 90 L	A=1,3					
0025	90	E1(IA)=.	JLOCII.IA)					
0026		DG 30 J	∗IP1,NCH					
0027		lsum=c						
0028		.00 91 E	A=1,3					
0029		JL([A]=	JLCC(J,IA)	N				
0030		DO 91 1	B#1,3		•			
0031	91	16(11()	A]-EQ-11(I	0)) 15UM=15UM+	1			
0032		IFIISUM	.GT.1) GOT(	0 30				
0033		CALL LN	KCNTEMZ1.J.	(PASS) .				
C034		IFILPAS	5.EQ.1) GO	ra 30	1			
0035		CALL HO	BCK(J,MZI,)	NR + NCOL + KOBI		×		
0036		IFIMO8.	EC.0) GOTO	30				
0037		JSTORE	108L,1)=1					
0038		JSTORFE	IDBL+2}=J					
0039		KCUN1#K	UUNI+L					
0040		TELKUUN	1.61.53 CA	LL CHAINI				
0041	c	CALL CB	LAUDEJĮŅILI	CUL PRE LAREI				
0043	C FIL	100-1						
0042		DO 20 1						
0043		16/ 10/20		14 AND. (080HE	1. TAL. NE. 143	6010 20		
0044		CALL NY	E ILI / 1081 .	IA. IROWI				
0045	20	CONTINU						
0040	20	LUNIINU	6 .1.400011.					
0041		TON - TO	5151 CL-11					
0048	20	CONTINU	5 C					
0044	30	CONTINU	6 . E .					
0050	10	109/ = 10						
0052		1611081	GF.KSTOPE	CALL CHAINI				
0053		NCHEINA						
0054		NSTORF=	лсн					
0055		RETURN						

# TABLE LVI (Continued)

FCRTRAN	IV	G LEVEL	1, HOD 4	OBLADD	DATE = 70199	08/35/08	PAGE COOL
0001					1		
0001			INDI LCIT INTE	CF0=2{[-k]	•		
0002			COMMON/OBL/JL	OC(20.3).JOROW(20.15	.JRDN(20.15)		
0004			DIMENSION MYS	15,201,MZ(15,20)			
0005			IBASE=JLOC(1,	1)			
0006			00 10 IA=1.NR				
0007			IF(JOROW(1,IA	J.NE-IAJ GOTC 11			· ·
0008			NE=MY(IA NCOL	<b>)</b>			
0009			DD 20 18=1,NC	EQ. IBASEN COTC PI			
0010		20	CONTINUE				
0012		21	IX=18-1				
0013			IY=18+1				
0014			1F(18.EQ.1) 1	X=NE			
0015			1F(18.EQ.NE)	14=1			
0016	1		IF(MY(IA,IX).	EQ. JLOC(1,2) .AND.MY(	IA, IYJ.EQ. JLCCII, 333	GULO 30 GULO 30	
0017			TELNVILA, TV)	EQ.JLUCII:3/*AND*HTI	A.1X1.60.1100(1.31) G	0010 SC	
0019			TE(MY(1A.TY).	EQ. JL OC(1.2) - CR. MY(1	A.191.EO. JLOC(1.3)) G	010 50	
6620		30	DO 31 1C=1.NE				· .
0021			MZ{IA,IC}=MY(	IA,1C)		· • •	
0022		31	IFINZ(IA, IC).	EQ.JLOC(1,1)) HZ(IA,	IC)=1211		
0C23			K=NE+1	-			
0024			00 32 IC+K,NC				
0025		32	M2(1A,1L)=M11	14,101			
0028		40					
0028			00 41 1C=1.NE				
0029			MZ(IA.ICOL)=M	Y([A,]G)			
0030			ICCL=ICOL+1				
0031			IFIICOL.EQ.IX	+1) GOTO 42	• .		
0032			GOTO 41				
0033		42	MZ(1A, ICGL )=1	211	1		
0034		41	CONTINUS				
0036			00 43 IC#ICOL	-NCOL			
0037		43	MZ(1A.1C)=0		•		
0038			MZ(IA, NCOL)=N	E+1			•
0C35			GOTO 10				· .
0040		50	1 X=18				
0041			GOTO 40	<b>.</b> .			
0042		12	DU 12 10-1-NV	UL 14.14.1			
0043		10	CONTINUE	INTIDI		•	
0045			RETURN				
0046			END				
					1		
						1	
					04TC - 70100	04/75/09	84CE 0001
FORTRAN	1 4	G LEVEL	1, KLD 4	HAFILL	UATE - 10199		FAGE COOL
0001			SUBROUTINE MX	FILL(1.1A.1RDb)		,	
0002			IPPLICIT INTE	GER+211-N)			
0003			COMMON/ALL/HI	15,151,NB,NL,NR,NCOL	NK, NCH, NELEM, NSTORE		
0004			COMMON/DBLMX/	M2(15,15),MZ1(15,15)	+MZ2(15,15)+MZ3(15,15	J,MZ4(15,15)	
		4	M25(15,15), M25(15,15), M25(15,15),	MX1100,15,151			
0005			NE#M211A#NUUL	,			
6000			1E(#7/14.18).	FC-1211) GOTO 40			
0008		30	CONTINUE				
0009		40	ICCL=1				
001C			ISTOP=18				
0011		41	MX11,1ROW,ICO	L)=MZ(1A,18)			
0012			ICOL=ICOL+1				
0013			18=18+1	10-1			
0014			16/18-61-NE	10-1 9) COTO 50	1		
0015			COTO 41	ri 3010 34			
0010		-50	00 51 18=1COL	.NCOL			
0018		51	MX(1, IROW, 18)	=HZ(IA, 18)			
0019			IROW=IROW+1				
0020			RETURN				
CC21			END				



Figure 33. Flow Chart for Subroutine DBL2

# TABLE LVII

# SUBROUTINES LNKCNT AND MOBCK

FORTRAN	IA (	5 (	EVEL	1,	MOD	4		MOBCK		DATE =	70199	C8/35/08	P	AGE 0001
0001				ŞU	ROU	TIN	E MOBCKE	N. H. AR. NCO	L, KOB)					
0002		ζ.		LH	PL 10	11	INTEGER*	2(1-N)		10044.30	• •			
0003				011	IF NS	100	H(15.15	) 1	(20113)	, JKUW1 20, I				
0004		(	C CHE	CK	NLL	RON	S CONTAL	NING COUBL	E JOINT	TO INSURE	THAT NO	THO ROWS WHICH W	LL	
		(	CON	ΪA I I	I ON	LY	FOUR LIN	KS, HAVE T	HREE LI	NKS IN COM	MON.		-	
0005				LI	ijra	CIN	111							
0007				1.2	JLU 	C ( N	,21							
0008				NR	1=1	8-1								
0005				DO	10	1=1	NR#1							
0010				IF	JRO	WEN	.1).NE.[	) GOTO 10						
0011		(			UMB	ER	OF LINKS	EXCLUSIVE	OFLI					, i i i i i i i i i i i i i i i i i i i
0012				15	JM = 0									
0013				DO	20	J=1	NE1							
0014			20	IF	HCE	-1)	.LT.1000	.AND.HII.J	1.NE.L1	) ISUM=1SU	H+1			
0015				11	150		1.41 601	0 10						
0017				ĎŐ	30	Ĵ=1	P1,NR							
0018				1 F	JRO	WEN	JJ.NE.J	) GOTO 3C						
		(	: cou	NT	UNB	ER	OF LINKS	EXCLUSIVE	OF L1					
0019				NE.	2=146	<b>J</b> , N	CUL }	•						
0020				00	40	K=1	• NE 2							
0022			40	IF	INIJ	ιKĴ	LT.1000	.AND.M(J.K	) .NE .L 1	] ISUM=ESU	H+1			
0023				1 F	ISU	#. G	T.4) GOT	0 30	_					
0024		0	: DET	ERM	INE	16	THE TWO	LOOPS HAVE	MORE T	HAN THE LL	NKS IN C	CAHON		
0024				13	50 50	K.s. 1	AFT .							
0026				1F	HEI	,к I	.GT.1000	.OR.M(1,K)	.EQ.L1)	GOTO 50				
0027				DO	60	L=1	,NE2							
0028			60	IF	MEL	9K)	.EQ.M(J,	LIJ ISUM#E	SUN+L -	'n				
0029			50	LU HO	48 # # # 1 = C	ŲΕ								
0031				IF	i i su	M . G	T.2) RET	URN						
0032			30	CO	AT EN	UE		• ,						
0033			10	CO	NTEN	UE								
0034				RE	1118 V 1 = T									
0036				ENI	)					*				
												and the second		
FOR TRAN														
	EV C	; I	LEVEL	1,	×00	4		LNKCNT		CATE -	70199	08/35/08	P	AGE 0001
0001	1V (	5 1	LEVEL	1, Su	¥C0	4 171 A	E INKENT	LNKCNT	\$1	CATE -	70199	08/35/08	P	AGE OOUL
0001	IV (	5 1	LEVEL	l, Sui	FCC ROL	4 111 N	E LNKCNT Integer+	LNKCNT [MZ+L+IPAS 2(1-N)	S)	CATE -	70199	00/35/08	P	AGE 0001
0001 0002 0003	IV (	5 1	LEVEL	l, SU( IP CO	FCC ROL PLIC	4 111 11 1/AL	E LNKCNT Integer≠ L/M(15,1	LNKCNT (MZ+L+IPAS 2(1-N) 5]+NB+NL+N	S) Rancola	CATE =	70199 EM, NSTOR	08/35/C8 E	P	AGE 0001
0001 0002 0003 0004	IV (	5 1	LEVEL	1, SU( IF CO(	MCC IROL PLIC IMON	4 111 17 17 17 17 10 10 10 10	E LNKCNT INTEGER+ L/M(15,1 L/JLOC(2	LNKCNT [MZ+L+IPAS 2(1-N) 5]+NB+NL+N 0+3]+JDRCb	S) R., NCOL , (20,15)	CATE - NK, NCH, NEL , JROW (20,1	70199 EM, NSTOR 5)	00/35/C8 E	β	AGE 0001
0001 0002 0003 0004 00C5	1V (	5 4	LEVEL	1, SU( IF CO) O(	FCC ROL PLIC MON MON	4 111 17 17 17 17 17 17 17 17 17 17 17 17	E LNKCNT INTEGER+ L/M(15,1 L/JLOC(2 M2(15,1	LNKCNT (M2+L+IPAS 2(1-N) 51+N8+NL+N 0+31+JDRCb 51	S] R., NC QL , { 20,15}	CATE - NK, NCH, NEL , JROW (20,1	70199 EM, NSTOR 5)	08/35/C8 E	P	AGE 0001
0001 0002 0003 0004 0005 0006 0007	1v (	6 1	LËVEL	1, SU( IF CO) CO) D1 L1 L2	MCC IROL INON IMON IMON IENS JLC	4 111 1/AL 1/DB 10N 101 101	E LNKCNT INTEGER+ L/M(15,1 L/JLOC(2 M2(15,1 ,1) ,2)	LNKCNT [MZ+L+IPAS 2(1-N) 5]+NB+NL+N 0+3]+JDRCh 5}	S] R, NCOL, {20,15}	CATE = NK,NCH,NEL ,JRQW{20,1	70199 EM, NSTOR 5)	08/35/C8 E	P	AGE 0001
0001 0002 0003 0004 0005 0006 0007 0007	1v (	i 1	LEVEL	1, SU( IF CO) D1 L1 L2 L3	MCC IROL PLIC IMON IMON IENS JLC JLC	4 111 1/AL 1/DB 10N 10(L 10(L	E LNKCNT INTEGER+ L/M(15,1 L/JLOG(2 M2(15,1 ,1) ,2) ,3)	LNKCNT (M2+L+IPAS 2(1-N) 5]+N8+NL+N 0+3]+JDRCh 5}	S] R,NCQL, (20,15)	CATE = NK,NCH,NEL ,JRQW(20,1	70199 EM, NSTOR 5)	08/35/C8 E	Ρ	AGE 0001
0001 0002 0003 0004 0005 0006 0007 0008 0009	1v (	5 1	LËVEL	1, SU( IP CO( D1) L1: L2: L3: IP/	HCC IROL INON IMON IENS JLC JLC	4 111 1/AL 1/DB 10N C(L C(L C(L C)	E LNKCNT INTEGER+ L/H(15,1 L/JLOC(2 MZ(15,1 ,1) ,2) ,3)	LNKCNT (MZ+L+IPAS 2(1-N) 5]+NB+NL+N 0,3]+JDRCh 5}	S] R, ACOL, (20,15)	CATE - NK,NCH,NEL ,JROW(20,1	70199 EM, NSTOR 5)	08/35/C8 E	Ρ	AGE 0001
0001 0002 0003 0004 0005 0006 0007 0008 0009 0010	1v (		LEVEL	1, SU( IF CO CO DI L1: L2: L3: IP/ DO I6	HCC IROL INON IMON IMON IMON IMON IMON IMON INON IN	4 111 1/AL 1/DB 101 101 101 101 101 101 101	E LNKCNT INTEGER+ L/M(15,1 L/JLOC(2 MZ(15,1 ,1) ,2) ,3) ,NR COL1-GI	LNKCNT (M2,L,IPAS 2(1-N) 5),N8,NL,N 0,3),JDRCb 5)	S) R, ACQL, (20,15)	CATE - NK,NCH,NEL ,JROW(20,1	70199 EN, NSTOR 5)	08/35/08 E	P	AGE 0041
0001 0002 0003 0004 00C5 00C6 0007 00C8 0009 0010 0011 0012	¥¥ (	i 1	LEVEL	1, SU( IF CO( D1) L1 <sup>2</sup> L2 <sup>2</sup> L3 <sup>2</sup> IP/ D0 IF	HCC IROL PLIC IMON IMON IMON IMON IMON IMON IMON IMON	4 111 120 100 100 100 100 100 100 100 100	E LNKCNT INTEGER+ L/M(15,1 L/JLOC(2 M2(15,1 ,1) ,2) ,2) ,3) ,NR COL).GT. COL)	LNKCNT (MZ,L,IPAS 2(1-N) 51,NB,NL,N 0,31,JDRCb 5} NELE#+3} G	S) R,NCQL, (20,15) GT& 10	CATE - NK,NCH,NEL ,JROW(20,1	70199 EM, NSTOR 5)	08/35/C8	ρ	AGE 0001
0001 0002 0003 0004 00C5 00C6 0007 00C8 0009 0010 0011 0012 0013	<b>IV</b> (	5 1	LEVEL	1, SU( IF CO OI L1: L2: L3: IP/ DO IF NE: 1F	#CC BROL PLIC IMON IMON IENS JLC JLC ISS= 10 IMZ I INZ I	4 111 14 100 101 100 101 100 101 100	<pre>E LNKCNT INTEGER+ L/M(15,1 L/JLOC(2 MZ(15,1 ,1) ,2) ,3) ,NR COL).GT. COL) O GOTO</pre>	LNKCNT (MZ,0L,1PAS 2(1-N) 5),768,71,70 0,3),JDRCb 5) Nelep+3) G 10	S) R.NCQL, {20,15} GTQ 10	CATE - NK,NCH,NEL ,JROW(20,L	70199 EM, NSTOR 5)	00/35/C8	β	AGE 0041
0001 0002 0003 0004 0005 0006 0007 0008 0009 0010 0011 0011 0012 0013 0014	1v (	5 1	LEVEL	1, SU( IF CO OI DI L1: L2: L3: IP DO IF NE: IF	#CC IROL IROL INON INON ISS ISS ISS INC INC INC INC INC INC	4 111 17 17 17 17 17 10 10 10 10 10 10 10 10 11 11 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	E LNKCNT INTEGER+ L/M(15,1 L/JLOC(15,1 ,1) ,2) ,3) ,NR COL).GT. COL).GT. COL) O) GOTO	LNKCNT (MZ,L,IPAS 2(1-N) 5),AB,NL,N 0,3),JDRCb 5) NELE#+3) G 10	S) R, NCOL, ( 20,15) CT& 10	CATE - NK,NCH,NEL ,JROW(20,1	70199 EM, NSTOR 5)	00/35/C8	β	AGE 0041
0001 0002 0003 0004 0005 0006 0007 0008 0009 0010 0011 0012 0013 0014 0015 0014	1v (	6 1	LEVEL	1, SU( 1P CO( 01) L1: L2: L3: IP/ DO IF IS( JS)	MCC IROL IROL INON INON ILC ILC INC INC INC INC INC INC INC	4 111N 17AL 17AL 17AL 17AL 17AL 17AL 17AL 17AL	E LNKCNT INTEGER+ L/M(15,1 1) ,2) ,2) ,2) ,2) ,2) ,2) ,2) ,2) ,2) ,2	LNKCNT (MZ,L.,IPAS 2(1-N) 5), AB,AL,N 0,3), JDRCb 5) NELEP+3) G 10	S) R, NCOL, (20,13) GT& 10	CATE - NK,NCH,NEL ,JROW(20,1	70199 EM, NSTOR 5)	08/35/C8	β	AGE 0041
0001 0002 0003 0004 00C5 00C6 00C8 0009 0010 0011 0012 0013 0014 0015 0016 0017	1v (	6 1	LEVEL	1, SU( 17 01 12 10 17 10 17 15 15 15 15 15 15 15 15 15 15 15 15 15	MCC BROL PLIC MMON JLC JLC JLC ISS= 10 (M2( INE. JH=C JH=C JLC INE.	4 ITIN //AL //DB C(LL C(LL C(LL C(LL I)N E I)N E I)N E I)N I)N I)N I)N I I I I I I I I I I I I	E LNKCNT INTEGER+ L/JLOC(2 #2(15,1 ,1) ,2) ,3) ,NR COLJ.GT. COLJ.GT. COLJ 0) GOTO NE	LNKCNT (MZ,L.) [PAS 2(1-N) 5),AB,AL,N 0,3),JDRCh 5) NELEP+3) G 10	S) R, NCOL, (20,15) GTG 10	CATE - NK,NCH,NEL ,JROW(20,1	70199 EM, NSTOR 5)	08/35/08 E	β	AGE 0001
0001 0002 0003 0004 00C5 00C6 0007 00C8 0010 0011 0012 0013 0014 0015 0014 0017 0018		5 1	LË VË L	1, SU( 17 01 12 17 100 17 15 15 15 15 15 15 15 15 15 15	MCC BROL MON MON JLC JLC JLC ISS= IMZ ( IMZ ( IM	4 ITIN ITIN ITIN ITIN CIL IIIN CIL CIL IIIN CIL CIL CIL IIIN CIL CIL CIL CIL CIL CIL CIL CIL	E LNKCNT INTEGER+ L/H(15,1 ,2) ,3) ,3) ,73 ,73 ,73 ,73 ,73 ,73 ,73 ,73 ,73 ,73	LNKCNT (MZ,6L,IPAS 2(1-N) 5),NB,NL,N 0,3),JDRCb 5) NELEP+3) G 10 M-JSUM+1	S) R, NCOL, (20,15) GT& 10	CATE - NK,NCH,NEL ,JROW(20,1	70199 EM, NSTOR 5)	08/35/08 E	β	AGE 0001
0001 0002 0003 0004 0005 0007 0008 0000 0010 0011 0012 0013 0014 0015 0016 0018 0019	<b>IV (</b>	3 1	LEVEL	1, SUI 100 01 12 12 19 100 15 15 15 15 15 15 15 15 15 15 15 15 15	MCC BROLC BROLC MMONS JLC JLC JLC JLC IME IME JLC IME IME IME IME IME IME IME IME IME IME	4 NTIN 11T 12ALDON 12C(LL) 11,N. 1,J1L 1,0. 1,J1L	E LNKCNT INTEGER+ L/ALIS,L L/JLOC(2 W2(15,1 ,1) ,21 ,3) ,31 ,33 ,33 ,01 GOTO OJ GOTO NE 00001 JSU 1.000.4.5E	LNKCNT (MZ,L.IPAS 2(1-N) 5),NB,NL,N 0,3),JDRCb 5) NELEP+3) G 10 M-JSUM+1 G-L2.CR.K.	S) R, NCOL, (20,15) CTQ 10 Eq.L3)	CATE - NK, NCH, NEL , JROW ( 20, 1 ISUM-15UM+	70199 EM, NSTOR 5)	00/35/C8	β	AGE 0041
0001 0002 0004 0005 0007 0009 0010 0011 0012 0014 0015 0014 0015 0017 0019 0021	<b>IV</b> (	5 I	LEVEL	1, SUI 100 011 123 100 17 17 17 15 17 17 17	# CC           IR OLC           IR OLC           IMON           IMON           I MON           I JLC           I JLC <th>4 NTIN 11T L/ADBN 10C(LLL) 10C(LLL) 11,N. 11,N. 11,1 10,1</th> <th>E LNKCNT INTEGER+ L/JLOC(2 #2(15,1 ,1) ,2) ,3) ,7R COLJ.GT. COLJ.GT. COLJ O GOTO NE 0000J JSU L.OR.K.E Q.3J JSU M.LF.3J JSU</th> <th>LNKCNT (MZ,L.,IPAS 2(1-N) 5),JB,KL,N 0,3),JDRCb 5) NELEP+3) G 10 M=JSUM+1 Q.12.CR.K. M=JSUM+1 Q.12.CR.K.</th> <th>S) R, ACOL, (20,15) CTO 10 EQ.L3)</th> <th>CATE - NK, NCH, NEL , JROW (20,1 ISUM-15UM+</th> <th>70199 EM, NSTOR 5)</th> <th>08/35/C8 E</th> <th>β</th> <th>AGE 0041</th>	4 NTIN 11T L/ADBN 10C(LLL) 10C(LLL) 11,N. 11,N. 11,1 10,1	E LNKCNT INTEGER+ L/JLOC(2 #2(15,1 ,1) ,2) ,3) ,7R COLJ.GT. COLJ.GT. COLJ O GOTO NE 0000J JSU L.OR.K.E Q.3J JSU M.LF.3J JSU	LNKCNT (MZ,L.,IPAS 2(1-N) 5),JB,KL,N 0,3),JDRCb 5) NELEP+3) G 10 M=JSUM+1 Q.12.CR.K. M=JSUM+1 Q.12.CR.K.	S) R, ACOL, (20,15) CTO 10 EQ.L3)	CATE - NK, NCH, NEL , JROW (20,1 ISUM-15UM+	70199 EM, NSTOR 5)	08/35/C8 E	β	AGE 0041
0001 0002 0004 0005 0007 0009 0010 0011 0012 0013 0014 0015 0014 0015 0014 0017 0014 0017 0019 0020 0022	<b>EV (</b>	à 1	5	1, SUP COULL2: 10 L1: L2: 10 F SUP L1: L2: 10 F SUP L1: L2: 10 F L1: L2: 10 F L1: L2: 10 F L1: L2: 10 F L1: L2: L2: L2: L2: L2: L2: L2: L2: L2: L2	# CC           IR LICO           IR LICO           IMON           ISSO           ISSO           IME           IME           IME           IME           IME           IME           IME           INE           INE           INE           INE           INE	4 NTIN 111 124 120 120 120 120 120 120 120 120	É LNKCNT INTEGER+ L/JLOG(2 #2(15,1 ,1) ,2) ,3) ,NR COLJ.GT. COLJ O J GOTO NE 0000 J JSU 1.00.K.FE Q.3) JSU M.LE.33 EQ.1) RE	LNKCNT (MZ+L.IPAS 2(1-N) 5), AB, AL, N 0, 3), JDRCh 5) NELEP+3) G 10 M-JSUN+1 0, 12.CR-K- M-JSUN+1 IPASS=1 TURA	S) R, ACOL, (20,15) CTO 10 EQ.L3)	CATE - NK,NCH,NEL ,JROW{20,1	70199 EM, NSTOR 5)	08/35/C8 E	β	AGE 0001
0001 0002 0004 0004 0005 0007 0008 0010 0011 0012 0013 0015 0015 0016 0017 0018 0016 0017 0018 0016 0010 0020 0021 0023	<b>1 v</b> (	à 1	5 10	1 , SUP COUL L23 DOF IF IF IF IF IF IF COU IF IF IF IF IF IF IF IF IF IF	# CC BROLCO BROLCO BRONSO JLCO SJLCO	4 ITIN 1 TIN 1	E LNKCNT INTEGER+ L/JLOC(2 W2(15,1 ,1) ,2) ,3) ,NR COL).GT. COL) GOL).GT. COL) GOJ JSU LOR.K.E Q.3) JSU M.LE.3) EQ.1) RE	LNKCNT (MZ,6L,1PAS 2(1-N) 5),NB,NL,N 0,3),JDRCb 5) NELEP+3) G 10 M=JSUM+1 G_L2_CR+K- m=JSUM+1 IPASS=1 TUPA	S) R, NCOL, (20,15) GT& 10 EQ.L3)	CATE - NK, NCH, NEL , JROW (20,1 ISUM-15UM+	70199 EM, NSTOR 5)	08/35/08 E	β	AGE 0001
0001 0002 0004 0004 0005 0007 0008 0007 0010 0011 0012 0013 0014 0015 0016 0015 0016 0019 0010 0011 0012 0013 0014 0015 0016 0010 0010 0010 0010 0010 0002 0003 0004		à 1	5 10	1, SIP00011 L12 L19000000000000000000000000000000000000	# CO BROLO B	4 ITIN 17 I	E LNKCNT INTEGER+ L/ALIS,L L/JLOC12 W2(15,1 ,1) ,21 ,3) ,31 ,30 GOLJ.GT. COLJ.GT. COLJ.GT. COLJ.GT. COLJ.GT. COLJ.GT. COLJ.GT. COLJ.GT. COLJ.GT. COLJ.GT. RE 00001 JSU L.OR.K.E Q.33 JSU M.LE.33 EQ.1) RE	LNKCNT (MZ.6IPAS 2(1-N) 5), NB.N.N 0.3), JDRCb 5) NELEP+3) G 10 M=JSUM+1 G.12.CR.K. M=JSUM+1 IPASS=1 TURN	S) R, MCOL, (20,13) GTO 10 EQ.L3)	CATE - NK, NCH, NEL , JROW ( 20, 1 ISUM-15UM+	70199 EM, NSTOR 5)	00/35/C8 E	β	AGE 0001



Ł

**4**+--



Figure 34. Flow Charts for Subroutines LNKCNT and MOBCK

145

# VITA

Frederick Fenn Quist, Jr.

Candidate for the Degree of

Master of Science

#### Thesis: STRUCTURAL SYNTHESIS AND ANALYSIS OF KINEMATIC CHAINS USING PATH MATRICES

Major Field: Mechanical Engineering

Biographical:

- Personal Data: Born in Sault Ste. Marie, Michigan, September 29, 1942, the son of Col. and Mrs. Fred F. Quist.
- Education: Graduated from Wakefield High School, Arlington, Virginia, in June, 1960; received the Bachelor of Science degree from the United States Military Academy Academy, West Point, New York, in June, 1964; completed requirements for the Master of Science degree at Oklahoma State University in May, 1971.
- Professional Experience: U. S. Army Officer, 1964-1969; graduate research assistant, School of Mechanical Engineering, Oklahoma State University, 1969-1970.