Abstract—This primarily tutorial paper on the use of binary matrices in system modeling also includes new material related to the initial development of such matrices. The decomposition of binary matrices into levels such that all feedback is contained within the levels is illustrated. A method for developing a binary matrix en route to a structural model of a system is outlined. The development procedure partitions the matrix on the basis of supplied data entries. Then the interconnections between subsystems are added. This procedure permits transitivity to be used in developing the matrix.

I. INTRODUCTION

THIS PAPER reviews basic aspects related to matrices of system modeling that contain only entries of 0 or 1, called “binary” matrices. Some new material related to the development of such matrices is also presented. Binary matrices have great utility in the initial organizational phases of system modeling. Much is known about such matrices that does not seem to be widely used. The purpose of this paper is to present a mixture of old and new knowledge and to identify some key references in the belief that this will help open up the use of binary matrices to a considerably larger class of users.

II. PROPERTIES OF BINARY MATRICES

Binary matrices are useful because they can represent the presence or absence of a specified kind of relation between pairs of elements of a system, thereby opening up opportunities for structuring the system. Before the possibilities for system structuring are explored, a number of preliminary ideas will be presented that have utility in the more comprehensive discussion to follow.

Binary Relation

Consider any two elements of a system $s_i$ and $s_j$. Suppose it is possible to say either that $s_i$ and $s_j$ are related in a certain way, or they are not. That is, either

$$ s_i \not\sim s_j \quad (1) $$

or

$$ s_i \sim s_j \quad (2) $$

where the bar over $\sim$ is the negation. Then one can construct, for a system comprised of a set of $k$ elements $S = \{s_1, s_2, \ldots, s_k\}$, a binary matrix whose entry in position $(i,j)$ is 1 if (1) is true, and 0 if (2) is true. The matrix will be square. An example is

$$ M = \begin{bmatrix}
  s_1 & s_2 & s_3 & s_4 \\
  s_1 & 1 & 0 & 1 \\
  s_2 & 0 & 1 & 0 \\
  s_3 & 0 & 0 & 0 \\
  s_4 & 0 & 0 & 0 \\
\end{bmatrix} \quad (3) $$

Equation (3) then portrays the connections among the separate pairs of elements of the system.

A system graph may be constructed by allowing a vertex on the graph to represent a system element, and an edge joining two elements to represent a 1 in the matrix. In this way a graph of Fig. 1 is readily constructed from (3).

In the construction of the graph, it is seen that there is a certain asymmetry involved. For example, while $s_3 \sim s_4$ and $s_4 \sim s_3$ indicate that $s_3$ and $s_4$ have a kind of symmetric relationship, it is seen from the matrix that $s_2 \not\sim s_3$ but $s_3 \sim s_2$. Evidently, there is a directionality associated with the chosen relation $\sim$. This directionality may be indicated on the graph by associating an element from the vertical index set of the matrix in (3) with the first term in (1) and (2) and an element from the horizontal index set of the matrix in (3) with the last term in (1) and (2), as has already been done implicitly in the foregoing discussion. Then, if $s_i \sim s_j$, an arrow is placed on the graph from vertex $s_i$ to vertex $s_j$, showing the directional nature of the equation. With this change the graph of Fig. 1 is replaced by the graph in Fig. 2.

In Fig. 2 it appears irrelevant whether the arrows are added on the loops at vertexes $s_1$ and $s_2$, but the question arises, for example, as to why $s_1 \not\sim s_1$ but $s_3 \sim s_3$. In many instances in the applications, the relation $\sim$ is of such a nature and the system analysis is conditioned such that it does not matter whether one shows $s_i \sim s_i$ or $s_i \not\sim s_i$. Convenien for can often be the deciding factor. Therefore, this issue need be considered only in a specific context. The graphs portrayed in this way are called directed graphs, or digraphs. Everything that follows in this paper relates to digraphs and their matrices.

The matrix in (3) is not the only matrix in which the relations portrayed in that equation could have been shown.

---

Manuscript received February 21, 1972; revised April 12, 1973.

The author is with Battelle's Columbus Laboratories, Columbus, Ohio 43201.
The matrix was constructed in the manner given simply because of the natural sequence of subscripts 1, 2, 3, and 4 on the s. There is often no good basis for initial numbering of system elements, and it may be very useful to show the same information as in (3) in a binary matrix, but with a different ordering on the rows or columns or (usually) both. For this purpose, it is helpful to consider permutation matrices.

**Permutation Matrices**

The following matrix is called an *identity matrix*:

\[
I = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]  

Suppose it is desired to construct from this matrix a permutation matrix P such that the product MP would permute the columns of matrix M defined by (3). In particular, suppose it is desired that in M a) column 3 replaces column 2, b) column 4 replaces column 3, c) column 2 replaces column 4, and d) column 1 is unchanged. This can be done by first making the desired changes in the matrix I to obtain the matrix P. Thus

\[
P = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]  

Carrying out the operation MP using Boolean multiplication and addition, one obtains the result

\[
MP = \begin{bmatrix}
s_1 & s_2 & s_3 & s_4 \\
s_1 & 1 & 1 & 1 \\
s_2 & 0 & 1 & 0 \\
s_3 & 1 & 0 & 1 \\
s_4 & 0 & 1 & 0
\end{bmatrix}.
\]  

The matrix P has an inverse \(P^{-1}\). If this matrix is multiplied into MP, one obtains

\[
P^{-1}MP = \begin{bmatrix}
s_1 & s_2 & s_3 & s_4 \\
s_1 & 1 & 1 & 1 \\
s_2 & 0 & 1 & 0 \\
s_3 & 1 & 0 & 1 \\
s_4 & 0 & 1 & 0
\end{bmatrix}.
\]  

Comparison of (7) with (3) shows that the two matrices contain exactly the same information. However, the form of the matrices is different because of the two permutations. The operation \(P^{-1}MP\) has interchanged the rows and columns of M in accordance with the desired permutation matrix P. It will be shown that particular forms of binary matrices are very useful in understanding system structure.

**Beginning with Several Sets**

Suppose that instead of initially having a single set of elements, one had several sets. Perhaps each of the sets had been thought to represent a subsystem, but it was felt that the particular groupings of these sets might not be adequate for system study. In this case, one can simply compile all the separate sets into a single set and construct a binary matrix for the whole set. This is, if one had sets \(S_1, S_2, \ldots, S_n\), one could form the set union and let

\[
S = S_1 \cup S_2 \cup \cdots \cup S_n.
\]  

Mention of this possibility is made now for two reasons. First, one should not suppose that because there are several different sets available at the beginning what is said in this paper is not applicable. Second, in the following it will be shown that one of the products of analysis of binary matrices can be a partitioning of the original set into smaller sets, based on the information contained in the binary relation matrix. Thus one may give up the original partition represented by \(S_1, S_2, \ldots, S_n\) in hopes of obtaining a superior partition later. This would correspond to the reorganization of a system.

**Example of a Partition**

Suppose a binary matrix has been found for a system as follows (this example is based on one of Steward’s [1]):

\[
M_1 = \begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
2 & 0 & 1 & 0 & 0 & 1 & 0 \\
3 & 0 & 0 & 1 & 0 & 0 & 0 \\
4 & 1 & 0 & 0 & 1 & 0 & 0 \\
5 & 0 & 0 & 0 & 1 & 0 & 0 \\
6 & 0 & 1 & 1 & 0 & 0 & 1 \\
7 & 1 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}.
\]  

where the designation s has been dropped and the elements are represented by numbers. Examination of (9) reveals no particular regularities other than that all elements on the main diagonal are 1, that is, every element in the system would have a loop in its graph.

Suppose, however, that a row and column permutation is applied which takes the matrix in (9) into

\[
M_2 = \begin{bmatrix}
1 & 3 & 7 & 5 & 4 & 2 & 6 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 \\
3 & 0 & 1 & 1 & 0 & 0 & 0 \\
7 & 1 & 0 & 1 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 1 & 0 & 0 \\
4 & 1 & 0 & 0 & 1 & 0 & 0 \\
2 & 0 & 0 & 0 & 1 & 0 & 1 \\
6 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}.
\]  

Careful inspection of (10) shows that this matrix can be partitioned into the following form, where each symbol represents a submatrix, and 0 represents a submatrix that is filled with 0's:

\[
M_2 = \begin{bmatrix}
C_1 & 0 & 0 & 0 \\
C_{21} & C_2 & 0 & 0 \\
C_{31} & C_{32} & C_3 & 0 \\
C_{41} & C_{42} & C_{43} & C_4
\end{bmatrix}.
\]  

The precise way in which this is done is shown in Fig. 3. The heavy lines surrounding the diagonal submatrices in Fig. 3 indicate the forming of the partition by taking each
smallest square submatrix possible that can be formed along the main diagonal, while allowing only 0’s to the right of each diagonal submatrix. A matrix in this form is called block triangular.

Comparison of (11) and Fig. 3 permits direct identification of the submatrices. For example, the diagonal submatrices are

\[
C_1 = \begin{bmatrix}
1 & 1 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1
\end{bmatrix}, \quad C_2 = \begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix}, \quad C_3 = \begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix}, \quad C_4 = \begin{bmatrix}
1 & 1 \\
1 & 1
\end{bmatrix}.
\]

The subsystems \(C_1^*, C_2^*, C_3^*,\) and \(C_4^*\) represented by these four matrices are called the constituents of the system. To see why, one can construct the graph of the system from Fig. 3, which will appear as shown in Fig. 4.

In Fig. 4, the four constituents are circled. It is notable that there can be only one-way connections between constituents, while two-way connections are possible within a constituent. The partition thus enables one to think of the system as a set of four subsystems whose graph appears in Fig. 5. This graph is a condensation of the graph of Fig. 4 and has the following binary matrix:

\[
M' = \begin{bmatrix}
C_1^* & C_2^* & C_3^* & C_4^* \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0
\end{bmatrix}.
\]

Fig. 5 can be very useful if one is seeking to make interpretations concerning the system. It shows, for example, that \(C_3^*\) and \(C_4^*\) are both related to \(C_1^*\), but \(C_3^*\) and \(C_4^*\) are independent, as are \(C_3^*\) and \(C_2^*\). In a similar manner, \(C_3^*\) is related to \(C_2^*\). Among other uses, for example, one can apply this type of analysis to determine the order in which a set of equations can be solved. For details of this particular application, one may refer to Steward [1], [2] or to several other references [3]-[5].

The essential fact required to develop this representation is the partition on the set. That is, one can write

\[
S = \{1, 3, 7; 5; 4; 2, 6\}
\]

where each term identified by the overbar is called a block of the partition [6]. Evidently, one block represents one subsystem, or one constituent, when formed in the manner illustrated. Also, it is worth noting that there is a submatrix corresponding to each pair \((B_i, B_j)\) of blocks in the partition, for a total of \(2^4 = 16\) submatrices.

**Cycles**

If, in a digraph, there is a path from a vertex back to itself, the path is called a cycle. Fig. 4 illustrates several cycles. Cycles that touch only one vertex are called loops. Loops are generally the least interesting cycles. In \(C_1^*\) of Fig. 4, one notices several cycles, three of which are loops. There is a cycle that connects vertices 1 and 7. It is important to observe that the only cycles in Fig. 4 are found within constituents, rather than between two or more constituents. There are no cycles in the graph of Fig. 5.

The term “feedback” is often used to describe cycles. All feedback in Fig. 4 is contained within constituents. The partition given in (14) separates the system into constituents such that there is no feedback between constituents.

**Two Special Kinds of Systems**

Suppose that, not only is it possible to find a partition, but also that all of the submatrices other than those on the main diagonal (i.e., submatrices such as \(C_{21}, C_{31}, C_{32}\), etc. in (11)) are filled with 0’s. The corresponding matrix is called block diagonal. Then there is no connection between constituents, and the system is described as completely decomposable. If one is working with a system that is representable by such a matrix and makes the assumption that he can deal with it as though it consisted of independent subsystems, he is assuming that all those off-diagonal submatrices are 0.

Suppose that it is possible to find a partition such that all submatrices lying on the main diagonal are filled with 0’s and all entries to the right of this set of submatrices are 0, but the submatrices to the left of the main diagonal are not all filled with 0’s. The matrix is then block triangular. (This occurred, for example, in the matrix \(M'\) in (13).) The system
represented by such a matrix is called a hierarchical system, or hierarchy, and each block of the partition may contain several constituents, all of which lie at the same level of the hierarchy.

Hierarchies

Referring again to $M'$ of (13), one observes that it can be partitioned as shown in Fig. 6, taking the largest possible submatrices on the main diagonal which are filled with 0's and having no 1's to the right. The partition used in Fig. 6 is 1,2; 3,4. The two blocks of this partition represent the two levels of the hierarchy. The elements contained in a block represent the elements at a particular level in the hierarchy. Fig. 5 is, therefore, described as a two-level hierarchy.

Transitivity

It may happen that, for some system, it is specified that the relation $R$ is transitive. That is, if $s_i R s_j$ and $s_j R s_k$, then, necessarily, $s_i R s_k$. If this transitivity condition is always satisfied and if, in addition, it is true that if $s_i R s_j$ and $s_j R s_i$, i.e., asymmetry applies, then a considerable amount of inference concerning structure is often possible. Moreover, the graph of a matrix to which these conditions apply will always be a hierarchy.

From the foregoing, one can then anticipate that the matrix for a hierarchy will either be completely decomposable, in which case there will be only one level in the hierarchy, or else the matrix will be decomposable into more than one level.

If the asymmetry condition is violated but transitivity still applies, the digraph is no longer hierarchical, but contains feedback. As is illustrated in Fig. 4, decomposition may be possible with feedback present.

There is a transitive asymmetric matrix corresponding to every hierarchical digraph, and there is a transitive matrix corresponding to every feedback digraph. Moreover, as will be seen, it is possible to define levels for a feedback digraph.

Powers of Matrices [7]

For decomposability to be possible, there must be at least one pair of vertexes on the graph such that one vertex cannot be reached by a path from the other. If this condition is not satisfied, every vertex is reachable from every other, and the graph is described as strongly connected. The possibility of decomposition can be determined from the reachability matrix of a directed graph. This matrix is transitive.

A method for finding such a matrix, given a digraph, will now be described.

Suppose a digraph is available. Let a matrix $N_i$ be formed that contains a 1 in position $i, j$ if and only if there is an edge directed from $i$ to $j$ or else $i = j$. Because of the latter condition, necessarily, $I + N_i = N_i$, where the addition is Boolean.

Let $N_k$ be the matrix obtained by raising $N_i$ to the $k$th power through Boolean operations. Then, necessarily, $N_k \geq N_{k-1}$ for any integer $k$ greater than 2, where the inequality sign is applied to all elements in corresponding positions in the two matrices. (This inequality can be verified by substituting $I + N_i$ in place of $N_i$ in the process of raising to powers and carrying out the arithmetic.) Wherever there is a 1 in matrix $N_k$, it means that element $j$ is reachable by a directed path from element $i$. For a finite graph containing $p$ elements, the longest possible nonredundant path can have length $p - 1$. Thus $N_i^{p-1} = N_p$ for any integer $k$ greater than $p - 1$. Thus the matrix $N_i^{p-1} = N_p$ is called the reachability matrix of the graph. The system will be undecomposable if the reachability matrix is the universal matrix, i.e., if it is completely filled with 1's.

For example, the reachability matrix of the undecomposable subsystem represented by $C_i$ of (12) is the universal matrix, while the reachability matrix of the condensed graph described by $M'$ in (13) is the same as $M' + I$.

The reachability matrix contains a 1 for every path in the digraph. It is a basis for construction of a digraph, as will now be illustrated.

III. CONSTRUCTING A DIGRAPH FROM A REACHABILITY MATRIX

In this section, a general scheme for constructing a digraph from a reachability matrix is presented. Methods were given previously for constructing the digraph for certain special cases [8]. The method given here will apply to any square transitive binary matrix with equal indexing by set $S$. Such digraphs may contain feedback. In spite of the presence of feedback, it is possible to define levels. Every element $s$ in a transitive matrix has a reachability set $R(s)$ consisting of all elements of $s$ lying on paths that can be originated from $s$ and an antecedent set $A(s)$ consisting of all elements of $s$ lying on paths that include $s$ but are not originated from $s$. By the definitions, the element $s$ is included in both sets. An element $r$ will be in the reachability set of $s$ if and only if $s R r$. An element $r$ will be in the antecedent set of $s$ if and only if $r R s$.

Given the reachability matrix, any element $s$ for which $R(s)A(s) = R(s)$ is defined as a member of the top-level set for that matrix, and lies in the top level on the digraph. With this definition, given any two elements $s_i$ and $s_j$ in the top-level set, symmetry applies in the relations among elements at the same level, i.e., either a) $s_i R_k s_j$, or) $s_j R_k s_i$. That is, any two elements at the same level are either not connected to each other or else there are two-way connections between the two elements. Moreover, the condition $R(s)A(s) = R(s)$ ensures that all connections from $s$ to another element are at the same level as $s$, while all connections from another element to $s$ are either at the same level or at a lower level.
Once the top-level set is identified for a matrix, the elements in the top-level set and all representations of connections to them can be removed from the matrix, leaving a transitive submatrix which will also have a top-level set. The top-level set of the submatrix will be the second-level set of the matrix. Proceeding in this manner, all the levels of the digraph can be identified. To each level in the digraph, there corresponds a square submatrix indexed by the top-level set of the matrix or some submatrix found in the manner described. The matrix can be re-arranged to show the separate blocks of the partition formed by the sets at the various levels. Each block represents at least one system constituent.

The following example will illustrate the development of a digraph from a reachability matrix. The matrix is given in Fig. 7. To make the example completely clear, the entire collection of reachability sets and antecedent sets will be written from inspection of the matrix. The reachability set for element 1 is found by inspecting row 1 of the matrix. Every 1 in row 1 corresponds to a column index, and every such column index will be in the reachability set of element 1. In this case, the only entry of 1 is in column 1, thus \( R(1) = 1 \). To find the antecedent set of element 1, inspect column 1. To every entry of 1 in column 1, there is a corresponding row index, and the set of such row indexes is the antecedent set of element 1. By inspection, \( A(1) = \{1,3,4,7,8,9,11,12,13\} \). Following this principle, Table I is prepared, which shows the reachability sets and antecedent sets for every element of the matrix of Fig. 7, along with the set product \( R(s)A(s) \) of the reachability set and the antecedent set.

It is seen by inspection of Table I that the only elements for which the set product is equal to the reachability set are elements 1 and 2, hence these two elements are the top-level elements for the matrix of Fig. 5. Elements 1 and 2 can then be eliminated completely from Table I by removing all reference to these two elements in all four columns in Table I. If one were doing this manually, he could simply delete those elements from Table I and avoid preparation of a new table. For purposes of this discussion, Table II shows the result of such deletion.

Table II shows that the set product is equal to the reachability set only for elements 5 and 6, thus these elements are the top-level set for the reduced list, and hence are the second-level set for the matrix of Fig. 7. If elements 5 and 6 are deleted throughout Table II, one arrives at Table III. Table III reveals that the third-level set for the matrix consists of elements 4, 9, and 10 since the set products for these elements are equal to the corresponding reachability sets. Deleting reference to these three elements in Table III, one arrives at Table IV. Inspection of Table IV shows that the fourth-level set for the matrix of Fig. 7 consists of the elements 3, 8, 13, 14, and 15. Deleting references to these elements in Table IV leads to Table V. From Table V, it is clear that the only element at the fifth level is element 7. Deleting all reference to it in Table V leads to Table VI. Table VI shows that elements 11 and 12 lie at the sixth level.

The results of Tables I–VI indicate that the set of elements \( S \) that indexes the matrix of Fig. 7 can be partitioned into six blocks, each of which represents one level. That is,

\[
S = \{B_1; B_2; B_3; B_4; B_5; B_6\} = \{1,2; 5,6; 4,9,10; 3,8,13,14,15; 7; 11,12\}.
\]
It is clear from preceding discussions that constituents and levels have some similarities and some differences. A constituent is formed by taking the smallest possible diagonal submatrix such that there are no 1's to the right in the matrix, while a level in a hierarchy is identified by taking the largest possible diagonal submatrix that is filled with 0's (except on the main diagonal) and has no 1's to the right.

The more general definition of level illustrated by the present example shows that constituents can be defined within levels. For example, the submatrix defined by block 4,9,10 consists of two constituents that lie at the same level.

The submatrix defined by block 3,8,13,14,15 has four constituents at the same level. These remarks are illustrated in the arrangement of the matrix shown in Fig. 8.

While a given digraph has a unique reachability matrix, a given reachability matrix does not necessarily define a unique digraph. There may be several digraphs that have the same reachability matrix. They will all have the same vertexes, but may not have the same number of edges. If a
machine is to draw a digraph for a reachability matrix, it must have a way of determining which particular digraph to construct, i.e., which particular set of edges to draw.

Since any elements in the same constituent are necessarily mutually reachable because of the symmetry existing within a level, one can minimize the number of edges in a constituent having more than one element by electing to draw a constituent in circular form as indicated in Fig. 9.

Then each level consists of isolated vertexes and circles. Since any member of a constituent can represent it in connections between levels, a constituent can be thought of as an isolated vertex, insofar as such connections are concerned. This means that a procedure that would prescribe a way of developing a unique hierarchy could also be applied to develop the digraph of a matrix containing feedback. The element method described in [8] is such a procedure, and it has the advantage that the matrix need not represent a single (connected) hierarchy, but can represent several hierarchies.

Fig. 10 shows a digraph of the matrix in Fig. 8 that can be drawn based on the preceding discussion.

IV. SEMIDECOMPOSITION

If transitivity does not apply, the system may be undecomposable, because it may not be possible to construct a matrix partition of the type illustrated earlier, wherein there are no 1’s to the right of the diagonal submatrices. In such instances, the system may be semidecomposable. To illustrate this, a generalization of a partition called a set system [6] is introduced. Consider, for example, the non-transitive matrix

\[
M = \begin{bmatrix}
1 & 2 & 3 \\
1 & 1 & 1 \\
2 & 1 & 0 \\
3 & 1 & 1
\end{bmatrix}
\] (15)

Let the set system, the generalized partition, be \( \Gamma, 2; \Gamma, 3 \), and let this be thought of as having two blocks, \( B_1 = \Gamma, 2 \), and \( B_2 = \Gamma, 3 \). Using the principle that connections will be confined to the interior of constituents corresponding to blocks of the set system whenever possible, it is clear that the graph of Fig. 11 can be used to represent the connections given in the matrix of \( M \). This type of semidecomposition has been found useful in the study of structure of sequential machines [6]. In the solution of simultaneous equations, one interpretation of Fig. 11 is that if a value is assumed for \( x_1 \), then the value of \( x_3 \) is determined, and this in turn determines a value for \( x_2 \). In a more general interpretation, one can say that the connection from one constituent to another implies dependence, as was also true for the decomposable systems. From the point of view of facilitating modular interpretation of systems, both decomposition and semidecomposition appear to be valuable.

V. MATRIX CONSTRUCTION

Thus far all the matrices or their digraphs have been assumed to be given, but in the practical situation of problem solving, one is usually in a position of having neither a matrix or a digraph. Since a digraph can be formed from the reachability matrix, it would be helpful to have an efficient way to construct such a matrix.

Construction of a matrix is somewhat analogous to development of a set of equations for a complex system. This is itself a complex undertaking, and there are occasions when the only kinds of equations one can write are logic equations. Arrow [9] has called attention to the importance of logic equations in the analysis of social systems. Yet one does not find many instances in the literature where explicit use of logic equations is made in analyzing such systems. Sometimes one would expect that there would be a mixed set of continuum or interval equations and logic equations [10], but analyses of such systems are quite rare. One must suppose that the great difficulty of developing such systems of equations is a major inhibiting factor. It appears that there would be significant application for a method that would permit the direct synthesis of a system graph. Such a method would virtually enable one to begin with only a
mental concept of the important elements and types of relations characterizing a system and proceed to the development of a graph of the system without going through the intermediate step of developing and manipulating a large set of equations for the system. It may seem that such a process is virtually impossible, yet it turns out that it is quite possible to carry out a process that approaches this description. This process can be grossly described in terms of model-exchange isomorphisms.

Model-Exchange Isomorphisms

A model-exchange isomorphism (MEI) is a procedure whereby one type of model can be exchanged for another. For example, if one had a matrix that was considered to be a system model, but one desired to replace that matrix with a model consisting of a system graph, the MEI for this situation would consist of a procedure whereby a graph could be developed from the matrix. Section III presents an example of such an MEI, applicable to systems representable by digraphs.

To develop a digraph when no matrix is known, it is assumed at the outset that an intelligent individual or group has a mental model (possibly very fragile, and relatively disconnected) of some system and desires to arrive at a graphic model of the system. One would then require one or more MEIs to convert the mental model into the graphic model of the system.

In the scheme to be described, one can imagine that the MEI for going from the mental model to a system digraph is composed of an MEI for taking the mental model into a set of data points representing pair relations among system elements, an MEI for taking the data points into a matrix, and an MEI for taking the matrix into the system digraph. All three of these constituent MEIs make up the master MEI for synthesizing a system digraph from a mental model. Let us consider these three isomorphisms in sequence.

Mental Model to Data Points

It is assumed that the developer of the model is able to furnish to a computer a list of elements that are important in the system to be modeled. This list is represented by a set $S = \{s_1, s_2, \ldots, s_k\}$ containing $k$ members. It is further assumed that the developer can answer "yes" or "no" to queries of the form

$$\text{is } s_i \text{ R}_1 s_j?$$

(16)

put by the computer, where $R_1$ represents "subordinate to." In other words, the developer is able to determine from his own knowledge whether one element is or is not in a subordinate relation to another. If the answer is "yes," a 1 is entered at matrix location $(i,j)$ in the computer memory; otherwise, a 0 is entered.

Now, as answers to such questions evolve, it would be anticipated that the developer might be overwhelmed by the large number of such questions, or might even begin to give inconsistent answers, violating his own previous inputs. However, it is possible to attack both of these possibilities directly in the following way. The computer can be programmed to put only those questions that do not permit inconsistencies to be entered in the data, and to infer information based on rules of inference from past data. In this way, the burden on the developer is minimized by reducing the number of questions to be asked. Furthermore, any errors that are entered into the system are not compounded by further input errors. That is, the computer preserves faithfully the data entered by the user, and all of the consequences that the user might not have been able to anticipate. This is all that can be expected in model development. While no model development process can correct for errors in input judgmental data, it is possible to sustain consistency in the use of input judgmental data. This is made possible by presuming that transitivity applies.

Data Points to Reachability Matrix

The most complex part of the overall MEI is the portion for converting data points to a reachability matrix. Thus the particular MEI for converting data points to the matrix will be the subject of most of the ensuing discussion.

Partition on Elements: Suppose an arbitrary element $s_i$ is chosen from the initial set $S$, and the computer asks the developer for a series of paired comparisons of the type indicated in (16), which, collectively, achieve the following in the order given.

1) A subset $L(s_i)$ (not containing $s_i$) is formed from $S$, consisting of those elements of $S$ to which $s_i$ is subordinate. This subset is called the lift set of $s_i$.

2) A subset $D(s_i)$ is formed from those elements in $S - L(s_i) - s_i$ which are subordinate to $s_i$. This subset is called the drop set of $s_i$.

3) A subset $V(s_i) = S - L(s_i) - D(s_i) - s_i$ is formed, such that $s_i$ is not subordinate to any element in the vacancy set $V$, nor is any element in $V$ subordinate to $s_i$.

4) A subset $F(s_i)$ of $L(s_i)$, called the feedback set of $s_i$, is formed, consisting of those members of $L(s_i)$ that are subordinate to $s_i$.

The logic of such a partition of the set $S$ allows the unknown reachability matrix to be partitioned as shown in Fig. 12. The supplied data obtained in steps 1)-4) fill both the row and column indexed by element $s_i$. In addition to these rows and columns, the partition generates 13 submatrices. Of the 13 submatrices, 8 can be completely filled with inferred data stemming from the transitivity condition. Of the 5 remaining matrices, 3 are of the main diagonal type, and 2 are interconnection matrices.

The main diagonal matrices $M_{L-L}, M_{L-F}, M_{D-D}$, and $M_{V-V}$ can themselves be interpreted as reachability matrices for their respective index sets. Therefore, each of them can also be partitioned on some element contained in its index set.

Proceeding in this manner, all of the matrix entries can be obtained except those corresponding to the interconnection matrices such as $M_{F-L-L}$ and $M_{F-D}$ in Fig. 12, and others that may be designated by the partitions on elements in the main diagonal matrices.

Interconnection Matrix Development: Development of interconnection matrices is carried out in inverse order to their identification. The development of such matrices can be based upon an iterative algorithm, in which significant use can again be made of transitive inference. The theory of
Fig. 12. Partitioning unfilled matrix on one element.

development of these matrices is too extensive to be included in this paper. However, one may note that such matrices can always be thought of as interconnecting two hierarchies. Any feedback unit contained in either of the two subsystems can always be represented by a proxy element consisting of any member of the feedback unit.

VI. MATRIX TO SYSTEM GRAPH

When the reachability matrix has been completely filled, the procedure given earlier for constructing a digraph from its reachability matrix may be applied. The digraph itself will normally be replaced by a graphic portrayal showing the elements represented by the vertexes of the digraph. For example, the elements might be objectives, events, activities, motors, generators, radars, etc.

VII. CONCLUSION

Some of the basic properties of binary matrices that are of interest in structural modeling [7] have been illustrated. A method has been described that permits one to construct a reachability matrix with computer assistance and to convert it to a structural model of a system.

REFERENCES