AN ABSTRACT OF THE THESIS OF

Michael Shigeru Inoue for the Ph. D. in Industrial Engineering
(Name) (Degree) (Major)

Date thesis is presented September 2, 1966

Title SYSTEMS TABLEAU: AN INTEGRATED APPROACH TO SYSTEMS THEORY.

Abstract approved [redacted]
(Major professor)

Systems Tableau is suggested as a convenient tool for the integration of the three phases of systems theory: the synthesis of a model from the analysis of a system; the evaluation of the model; and the decision-making process for the design and control of the resulting system.

From a basic consideration of Man-Nature communications, several mathematical, biological, engineering, and management examples of systems models are examined to develop a unified definition of a system.

Logical, physical, mathematical, graphic, and computational requirements are postulated for the methodology of models for systems meeting the definition. These requirements are used to formulate the basic tableau as a hybrid of a mathematical mapping matrix and a graphical flowgraph that expresses the interrelationships among the components of a given system. Thus, a tableau is at once a

matrix and a network representation of the system.

The general (connecting), ordinal (dominating), and technological (directing) relations in the observation (phase) space are illustrated on tableaux for social, economic, management, and engineering examples. Related mathematics of relations are examined.

The relationships of these descriptive models to normative models are discussed as synthesis techniques. Orthogonalization of bases, parametric representations (in frequency space as probabilities and statistical distributions), and reductions in state (solution) space are methods introduced with examples in queueing, communication, and information models. In normative models, we are afforded some degrees of freedom expressed in terms of choice of alternatives. This decision requires at least an ordinal, if not cardinal, characterization of each alternative.

The ordinally normative models are based on comparatively quantifiable relations originally afforded by the uni-directional flow of time. Theories of Information, Algorithms, and Games were found useful in drawing valuable conclusions (decisions) from these models. Puzzles, games, Turing machines, and biological examples are discussed.

The cardinally normative models require decisions based on numerical values. A truly cardinal model must be cardinal resource-wise, time-wise, and information-wise. This interdependency of resources in phase space and information in state space, as functions of time expressible in frequency space, is the basis for the proposal of the Cardinal Utility Hypotheses. This concept allows the development of relations as peculiar Laplace-Z transform-pairs, with the utility of Information (usefulness of data for decision-making) serving as the Channel Capacity for a corresponding communication model.

The Principle of Optimality of Dynamic Programming was found most useful in Tableau, and its continuous counterpart of Maximum Principle is expected to take a respective place in the Calculus of Variations in Control Theory.

The relationship of the controllability and observability of a system and the diagonalization of its Tableau is also illustrated.

The linear models of the traditional Tableaux are reviewed and interpreted in the light of Systems Tableau Method. These include Quesney-Leontief Tableau Économique, Hellerman's Tableau, and Critical Path Scheduling Tableau. The obvious advantages afforded by the applications of Huggins' (and others) Signal Flowgraph techniques are briefly illustrated. Mention is made of a tableau-based computer program that will produce the dual network for Ford-Fulkerson's Minimum-cut-maximal-flow Method. A brief discussion of the future of Systems Theory concludes the treatise.

SYSTEMS TABLEAU:
AN INTEGRATED APPROACH TO SYSTEMS THEORY

by

MICHAEL SHIGERU INOUE

A THESIS

submitted to

OREGON STATE UNIVERSITY

in partial fulfillment of
the requirements for the
degree of

DOCTOR OF PHILOSOPHY

June 1967

APPROVED:

[redacted]

Associate Professor of Mechanical and Industrial Engineering

In Charge of Major

[redacted]

Head of Department of Mechanical and Industrial Engineering

[redacted]

Dean of Graduate School

Date thesis is presented September 2, 1966

Typed by Nancy Kerley

ACKNOWLEDGEMENTS

The author is deeply indebted to Dr. James L. Riggs for the unrestrained academic assistance and personal encouragement received especially during this most critical phase of graduate study.

Professor Louis Slegel's recommendation was instrumental in providing the writer the opportunity to attend the seminar on "Modern Systems Theory and Its Application to Large Scale Systems" held at the University of California at Los Angeles during the Summer 1966. The engineering extension of U.C.L.A. and Ford Foundation which provided the fund, Mrs. Ginni Meadows of the University Extension, and Professor Cornelius T. Leondes of the Department of Engineering should be acknowledged for this valuable experience.[1]

The advice and cooperation provided by the members of the graduate committee have been appreciated. Dr. Donald Guthrie of the Statistics Department who initiated the author into the study of dynamic programming and other aspects of operations research is thanked also for the use of his personal library. To Professor Thomas L. Yates, of the Computer Center, the author is grateful for his personal instruction and assistance in computer programming and operation as well as for his efforts in trying to arrange for

---

[1]References are made throughout the thesis to various materials presented at this seminar.

Systems and Procedures Association funds.

Professor Louis N. Stone, the Head of Electrical and Electronic Engineering Department should be thanked for foregoing personal inconveniences for the sake of this thesis. Other members of the Department of Mechanical and Industrial Engineering, especially Professor W. F. Engesser, K. N. Dervitsiotis and Howard Chamberlain should also be acknowledged.

Richard A. Gordon deserves special thanks for volunteering his valuable talent in drafting and for devoting time in proofreading and otherwise making the thesis possible. Miss Mary L. Shuhart has undertaken willingly the tedious task of compiling the bibliography and proofreading the manuscript.

Last but not least, the author is grateful to Mrs. Nancy Kerley who, for the second time in the author's career, has courageously undertaken the impossible task of expediently producing a superb work in typing. The cooperation of Type-Ink and Mrs. Muriel Davis has been appreciated in meeting the deadline.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF TABLES

SYSTEMS TABLEAU: AN INTEGRATED APPROACH TO SYSTEMS THEORY

I. INTRODUCATION

In Support of the Chosen Title

> To state the fact frankly is not to despair the future nor indict the past. The prudent heir takes careful inventory of his legacies, and gives a faithful accounting to those whom he owes an obligation of trust (Kennedy, 1961).

The seemingly pretentious choice of the title is not based on any claim to the birth of a new theory, or to the belittling of efforts made by many scientists, engineers, and philosophers in what we have come to call systems theory. Rather, it is meant to be an invitation to the readers for further testing of an integrated approach that has shown signs of promise in the limited applications we are about to present.

Systems Tableau is a tool conceived to facilitate the communication between various major techniques of systems theory. The most prominent ones are: group theory, matrix theory, network theory, mathematical programming, calculus of variations, and other related techniques gathered under the names of systems theory and Cybernetics. In addition, information theory and theory of algorithm are used as the general framework of modelling, and the stochastic nature of actual systems will be recognized whenever possible.

An earnest attempt has been made to investigate as many and varied fields and disciplines as possible in order to find common features underlying today's techniques, especially those emerging as the basis for a unified systems theory of tomorrow. Though there are several original techniques and concepts included in many phases of this presentation, no profound claims are intended in any one particular aspect of the systems theory. Rather, theorems and examples have been borrowed freely and deliberately from various fields, especially those that seemed typical and yet simple enough for Tableau applications. Whether this Tableau method can be applied as effectively for more sophisticated examples, or whether it should be replaced by a more advanced approach in the future, is not a question we attempt to answer. Our hope is that the Tableau approach will serve as an effective stepping stone for this body of knowledge so urgently needed to cope with the accelerating increase in the complexity of our society.

The use of the name "Tableau" will be justified by showing the close similarity between this Tableau and others such as: Tableau Économique, Simplex Tableau, CPS Tableau, and Hellerman's Tableau.

This treatise is divided roughly into three parts. The first section is an attempt to establish as much mathematical and philosophical basis to the concept of "systems" as we can afford without imposing a specialized restriction on the legitimate use of this term. The first chapter treats various concepts which formed the basis of today's systems theory, and tries to give "group theoretical" interpretations to these concepts. The second chapter examines essential properties of a "system" based on currently accepted definitions, and produces a unified definition which we shall adhere to for the remainder of the paper.

The second section purports the gradual development of the Tableau method as a synthesis, analysis, and decision-making tool in systems theory. The examples in this part come from a wide variety of disciplines including sociology, economics, engineering, physics, computers, games and puzzles, etc. In each case, the emphasis is placed on the various approaches used to cope with the complexity of the system. A complex system is usually made amenable to a systematic study through a series of aggregating processes. Homomorphic mapping to produce a finite number of discrete states; self-characterization techniques to replace subsystems by equivalent fixed components in larger systems study; synthesis of decisions based on information available in the reduced systems; development of algorithms for decision-making; formulation of optimal policy; and finally the recognition of recursive relations that will yield a rule for policy formulation are stages that our examples should render plausible, if not formally acceptable.

The last part is a brief historical recap of the highlights and our first and last attempt to define "systems theory."

Man and Systems

> ...by the mere quantitative differences in scale and complexity, modern large-scale systems have become qualitatively different from the smaller systems which have previously been studied; for the same reason, they have become qualitatively more like the two systems which have always been the prime objects of study of science: man himself and the physical world which constitutes his environment (Goode and Machol, 1957).

The purpose of this first chapter is to investigate man's relationships to what he calls systems.

A system above all is an abstract concept. Moreover, it is a man-made concept. Whether a system is classified as an artificial system or a natural system, it is the man who decides the elements that will comprise the system. What will or will not be included in the particular system is decided by a more or less precise recognition of relationships that would associate the particular element to the rest of items already in the system.

Depending on the relationships under consideration, a same element may be included in several systems. Mathematics may be considered as the cumulation of constant efforts expanded by the human race to abstractly express these relationships and the relationships between those of the different systems. This process can be perpetuated and used to find the cause-effect relationship of interest between elements in two different systems. In short, mathematics is a means that will allow a transition from the system in which the cause element (independent variable) is located [by tracing the relationships between systems (transformations)] until the chain of relationships will find the effected element of interest (dependent variable).

Two interesting consequences of this feature are: first, the birth of concept of an automation, and second, the use of mathematics and automaton in describing the communication systems. The representation of a system has been called a "model," and mathematics, par excellence, is a tool for building a "model." But it does not, by itself, indicate the aim for which the model was created, or how the system is to operate. The incorporation of the operating procedure into a model is the basic concept of an "automaton." Similarly, a model was found particularly suited to communicate information concerning the relationships to other individuals. In addition, a mathematical model could be used to model the "communication system" itself. When this modelling was applied to man's own communication system, his nervous system, this became the McCulloch-Pitts automaton model for neurology.

With this very broad context, we may say that there is no systems problems that can be solved without some mathematical technique. As the systems have become more and more complex with the advance of technology which enhanced our power of "association," the mathematics to manipulate these relationships had to become more sophisticated also.

When the complexity of the system has reached such an advanced level, however, to necessitate several specialists to solve a particular problem, with each representing a different branch of sciences and technology, we suddenly find that the advanced mathematics is too specialized to be used as a communication tool. It could no longer be used as a common language to describe the problems involved in the systems. If the problems could be described in such a manner, they would no longer be classified as a complex systems problem.

Thus, on order to discuss the problem itself, we have resorted to a more naive "black box" description resulting in a "block diagram" specifying the areas of study for each participant. The latter is essentially a diagram to show some suspected relationships between systems expressed in the form of "blocks" or "black boxes." We are essentially back to where we have originally started from: the study of more or less defined relationships.

If a truly basic study is to be undertaken to investigate systems theory, it seems therefore that we should start from the very beginning, investigating the basis for the mathematical systems themselves, namely the Set and Group Theories.

This chapter will therefore follow this cycle:

1. The development of simple mathematics of relations, namely the Set Theory.

2. The concept of Automaton: a system with an operating procedure.

3. The application of systems concepts to communication systems, including man's own body.

4. A formal study of the study of model-building: Group Theory and Transformations.

5. The development of the complex applied systems: Engineering and Management.

6. Finally, the need for a basic and systematic study of the systems themselves: Systems Science and Cybernetics.

Before undertaking the reading of the remainder of this chapter, our readers should be warned about the seemingly prodigal list of time-consuming definitions and trite-looking theorems awaiting them in the coming sections. Initially, the study was undertaken without this formal structure relying exclusively on commonly accepted meanings in interpreting terms used in systems theory. Unfortunately, the study quickly became an impossible task requiring a new set of interpretations each time we tried to penetrate a new field in which the systems theory is being used. The difficulty may be imagined easily by trying to distinguish descriptively such pairs of terms as homomorphism and isomorphism, set and group, identity mapping and permutation, endogenous and exogenous variables, system and environment, and others, without the help of a formal structure. The definitions and theorems are taken from Halmos (1958), Stephenson (1965), Courant and Hilbert (1953), and Mostow, Sampson, and Meywer (1963) with heavy preference on the last.

Mathematical Systems

> Car enfin, qu'est-ce que'l'homme dans la nature?
> Un néant à l'égard de l'infini, un tout a l'égard
> du néant, un milieu entre rien et tout...
>
> Il faut se connaître soi-même: quand cela ne servira
> pas à trouver le vrai, cela au moins sert à régler
> sa vie, et il n'y a rien de plus juste (Pascal, 1670).
>
> ---
>
> For, after all what is a man in Nature?
> A nothing with respect to the infinite, an all with
> respect to nothing, a medium between nothing and all...
>
> We must know ourselves: if this does not help to find
> the truth, at least it will serve to regulate our life,
> and there is nothing more just. (From Pensées of
> Blaise Pascal, a French philosopher born in Clermont-
> Ferrand, 1623)

Set Theory

The most fundamental concept of a "system" emerges from man's recognition of his own identity. Everything is either "he", the system, or what he can experience directly or indirectly, his environment. This philosophy is traceable at least to Pascal, if not as far as to Socrates. Each object a man encounters, therefore, can be classified into two sets: himself as a system, and nature as his environment.

Definition 1.1. Set. A "set" is a collection of certain objects, called the "elements" of the set. If S denotes a set, $x \in S$ will define x as an element of S.

After the man has learned to apply this dichotomy to all objects around him, he can then study the interaction between elements of the system and those of his environment. He recognizes the control exerted on him by his environment (e.g. "I am a victim of circumstances") and his capability to produce some change in his environment. Obviously this change is observable only when its feedback affects his system in some way (e.g. "to each action there is a reaction"). Thus, he has assumed the ability to associate with two objects (the system and its input) a third object (the output) related in some way (an operation). In engineering, we would call this a "black box concept." Mathematicians preferred to study it under the name "Group Theory" (or more broadly under "Set Theory").

From the identification of his own entity as a system, our man has learned to attribute similar identities to other subsets of his environment (a set).

Definition 1.2. Subset. A part of S is called a subset of S. Thus, it is possible to have: x e A e S, where x may be an element of the subset A of the set S.

The subset of particular interest is the system with which a man associates himself. The mere recognition of his own self as an entity in his mind means that there is a model of himself recorded as chemical or physical changes in his brain. This model is a collection of these changes which were created through an electro-chemical mapping operation. (Of course, we have yet to master any deep understanding of this process. See later discussion on page 24.)

Definition 1.3. Mapping. A mapping of a set S to a set T is a rule, or an operation (also called a function or an operation), which assigns to every element in S a definite element in T. If f denotes a mapping of S to T, then the element of T which f assigns to an element x of S is denoted by f(x) and we say that f sends (or maps) x into element f(x), an image of x.

Definition 1.4. One-to-one mapping. (Mostow, Sampson, and Meyor, 1963). A mapping f of a set S to a set T is called one-to-one iff (if and only if) the following conditions are satisfied:

1. For every element y in T there is an element x in S such that $f(x) = y$.

2. If x and x' are two different elements of S, then

$f(x) \neq f(x')$.

Definition 1.5. Identity mapping. $f(x) = x$ for all x in S is called an identity mapping.

Let us note several fine points. First, if f is a one-to-one mapping, then we could assign $f(x) = y \neq x$ such that x and y are both elements of the same set S. Such a mapping is usually called permutation and not identity mapping. Also let us point out a seemingly trivial but a vitally important consequence of our definition of one-to-one mapping:

Theorem 1.1. Existence of inverse mapping. To every one-to-one mapping f which sends a set S to T, there exists an inverse mapping f^{-1} which sends the set T to S, and which is also one-to-one.

Proof of theorem 1.1. If the mapping $f: S \to T$ is one-to-one, then we can define a mapping f^{-1} by assigning to each f(x) of T the element x of S that f sends to T. Now suppose that f^{-1} is not one-to-one. Then there must be at least one element f(x) of T for which f^{-1} will send an element z in S different from x. This contradicts the definition of f being one-to-one (Definition 1.4, Part 2), and x must be identical to z, or f^{-1} must also be one-to-one. We say that the mapping is unique $S \to T$ and $T \to S$. Q.E.D.

Definition 1.6. A X A. Let A and B be sets. A X B is the set of all ordered pairs (x, y) with x in A and y in B.

Using this newly defined set A X B, we notice that what we have been calling a mapping is really a subset of A X B. Our black box is a (x, y) relating our input x to our output y. This is akin to what we will later consider as "model synthesis." The next obvious step is to find the relation between the input and the black box so that the output may be found. This step which will correspond to our "model analysis" may be considered as the mapping of A X A into A. Obviously, this would be possible only if A is a set of all elements including both inputs and outputs.

Definition 1.7. Binary operation. A binary operation on set A is a mapping of A X A into A.

Example 1.1. The simplest example that will illustrate the binary operation is probably a binary number system. Let A be a set of two possible values: 0 and 1. Then A X A will be a set containing (0, 0), (0, 1), (1, 0), and (1, 1), the four possible permutations.

AND. One binary operation will map the first three elements of A X A into 0 and the last into 1. This will then become the Boolean AND (Intersection) operation.

OR. Another binary operation is an OR (union) relationship that maps the first element of A X A, namely (0, 0) into 0, and the remaining three elements into 1. Obviously, our example may also generate NAND and NOR in a similar manner. The result may be seen in the figure below.

A X A	AND	OR	NAND	NOR
(0, 0)	0	0	1	1
(0, 1)	0	1	1	0
(1, 0)	0	1	1	0
(1, 1)	1	1	0	0

Figure 1. 1. An example of binary operations.

Of course, a binary operation is not limited to a binary number system as it can be applied to any set A and its A X A. We shall use the symbol $\boxtimes$ to indicate a binary operation in any set A. However, we must be cautioned that $x \boxtimes y$ tells us nothing about $y \boxtimes x$.

Automaton Theory

Matrix representations. The term "binary" is used because of pairing of elements in set A (taken in specific order) to produce some new element of A. This suggests the representation of binary operations in matrix forms with rows and columns corresponding to elements of set A. If set A contained four elements a, b, c, and d, the general matrix representation of a binary operation would look like Figure 1. 2. a.

One binary operation that we are familiar with is that of addition. Figure 1. 2. b shows the + operation for our quadratic system. Replacing a, b, c, and d by 0, 1, 2, and 3 may make the meaning clearer.

$\otimes$	a	b	c	d
a	(a, a)	(a, b)	(a, c)	(a, d)
b	(b, a)	(b, b)	(b, c)	(b, d)
d	(c, a)	(c, b)	(c, c)	(c, d)
e	(d, a)	(d, b)	(d, c)	(d, d)

1.2.a.

+	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

1.2.b.

AND	0	1
0	0	0
1	0	1

1.2.c.

OR	0	1
0	0	1
1	1	1

1.2.d.

NAND	0	1
0	1	1
1	1	0

1.2.e.

NOR	0	1
0	1	0
1	0	0

1.2.f.

Figure 1.2. Matrix representations of binary operations.

In Figure 1.2.c, d, e, and f we have also illustrated the matrix representations for the binary operations of Figure 1.1.

State interpretation. Let us now consider a binary operation from a subjective point of view. We may, for example, put ourselves in a position of the first caveman who recognized himself as a system. Obviously such a genius deserves a name and we shall call him "Flintstone."

After our man Flintstone has identified every object as an element of his universe, a set of his entire knowledge, he has learned to take a subset and identify it as a "system." This system, for instance, might be his five fingers on one hand: a, b, c, d, and e. From this system of five elements he may formulate a quinary addition scheme as shown in Figure 1.3. Our readers are reminded that his "addition" is simply a binary operation relating two elements in his system to an element in the system. For example, a "thumb" (element a) and another "thumb" (a) gives an "index finger" (b), and so on.

Addition	a	b	c	d	e
a	b	c	d	e	a
b	c	d	e	a	b
c	d	e	a	b	c
d	e	a	b	c	d
e	a	b	c	d	e

Addition	1	2	3	4	5
1	2	3	4	5	1
2	3	4	5	1	2
3	4	5	1	2	3
4	5	1	2	3	4
5	1	2	3	4	5

Figure 1.3. Quinary addition.

Another interpretation can be considered by Flintstone as: "If I am in the state of 'thumb', and if I get another 'thumb', my next state will be an 'index finger.'" Thus, his five fingers are now constituting a "black box" with five possible internal states which can accept five types of inputs to yield five types of outputs.

Definition 1.8. Finite automaton. A finite automaton is a quintuple: $A = (I, 0, S, \lambda, \delta)$

where I is a finite set (the set of inputs)

0 is a finite set (the set of outputs)

S is a finite set (the set of internal states)

λ: $S \times I \rightarrow S$ is the next-state function, and

δ: $S \times I \rightarrow 0$ is the next-output function.

In this definition (Arbib, 1964), we are using the term "function" in preference to "mapping" (Definition 1.3). In the case of our man Flintstone who has just learned to count up to e, or 5 if you wish, the sets I, 0, and S are all identical and included five elements (a finite set) which could be the input, output, or the internal state. Nonetheless his binary operation (Addition = λ = δ) is a valid automaton.

Together with the concepts of "state" and "automaton," we have succeeded in introducing the concept of descrete time. An automaton at time t and internal state b will, upon the reception of an input c, produce output $\delta(b, c) = d$ (in the case of Flintstone) and a new internal state $\lambda(b, c) = d$ (in the case of Flintstone) at time t+1. Though we have already stated in Definition 1.3 that terms mapping, operation, function, and operator are synonymous, we prefer the word "function" in this instance to emphasize its connotation with "function of time."

Adic bases. What happens to our man Flintstone when he decides to add elements that cannot be accounted by his five fingers? One method is obviously that of increasing his system to include both hands and possibly both feet so that he can count up to 20 items. He could also initiate a concept of carry such that the thumb in his right hand will be worth one set of fingers on his left hand, and so on.

His concept of 168 might then be:

Big toe (right foot), Big toe (left foot), Index (right hand), Middle (left hand), or

$$1(5^3) + 1(5^2) + 2(5^1) + 3(5^0) = 168$$

Actually, this is less ridiculous possibly than the counting system adopted by some nations of today:

One (mile), One (yard), Two (feet), Three (Inches)

where inch stands for the length of the first segment of a thumb of some king; foot, the length of his foot, and so on, or

$$1(1,760 \times 3 \times 12) + 1(3 \times 12) + 2(12) + 3 \text{ feet.}$$

In each case, especially in the latter, it is obvious that a different mapping matrix must be used depending on which state we are in: right hand, inch, etc. . . . In the case of the latter, the same matrix (table) can no longer be used for mapping (computation). Inches and feet are like apples and oranges forming two different systems.

The expression of the type; $m = a_0 + a_1 b + \ldots + a_r b^r$ is called b-adic (diadic if b = 2, etc.), and b is called the base.

Number system.

> The system of the natural numbers 1, 2, 3, etc., is unquestionably the most important mathematical system. It is also the most familiar one, and the beginning student may wonder what there is to say about it that he does not already know. Yet that system has such an extraordinarily rich and complex structure that it is still the source of some of the deepest and most challenging problems of mathematics (Mostow, Sampson, and Meyer, 1963, p. 28).

In our decimal system the binary operations obey the same mapping rule regardless of the decimal position of the digits.

Therefore, if we were to build an automaton for any digit in the system, it would have the same next-state function as its next-output function. In other words, the table (matrix) to be used for the binary operation can also be used to indicate in which state the system is (tenth, hundredth, etc.). Contrast this with the British system where we will need one function (matrix) telling us how to add (or multiply, etc...) two elements in inches, and another function (matrix) that will tell us which state we are in and to which state we should move to next (say feet). The next state will have its own output function matrix (say an addition table in feet) and the state function matrix would have to be consulted again (to move up to yard).

Because of this ability of a metric system, we can map ten elements from our set into a subset, map ten of those subsets into another subset, and continue with this mapping until we have enumerated every element in the set regardless of how many elements there are, as long as they were finite in number.

Another concept worth noting is that of "threshold effect." In a decimal counting process, for example, the tenth digit will not change until the unit position has exhausted its elements. Until this saturation occurs in the lower digit, no change can be preceived in the higher digit.

These two concepts introduce the vital definition of isomorphism. However, before continuing with our discussion, let us turn our attention to the examples of "threshold" that exists outside of mathematical systems.

Communication Systems

> We are beginning to see that such important elements as the neurons, the atoms of the nervous complex of our body, do their work under much the same conditions as vacuum tubes, with their relatively small power supplied from outside by the circulation, and that the book-keeping which is most essential to describe their function is not only of energy. In short, the newer study of automata, whether in the metal or in the flesh, is a branch of communication engineering, and its cardinal notions are those of messages, amount of disturbance or "noise" -- a term taken over from the telephone engineer -- quantity of information, coding technique, and so on (Wiener, 1948).

External Memory

When Flintstone used his fingers to do simple computations (which are nothing more than a succession of mapping processes), his fingers in controlled position (pointing, folded, etc.) indicated the particular state his system was in. For example, folding his middle finger might have meant that he had three wives. Adding another one would put him in a new state of ring finger, or four. Thus, he has learned to code his internal state by causing a change in a physical object (his finger) that will retain some information.

Instead of his fingers he might decide to use markings on the wall of his cave. Three slashes might be used instead of his middle finger, and four instead of his ring finger. Considering himself as a system, we may look at this marking as an output of his system which is clearly observable as a physical change in his environment. When he wishes to retrieve his "external memory" (also called memory aid), all he needs to do is to look at the wall of his cave. Seeing the marking on the wall is an input to his system that has special coded information.

The reason why this is called "external" memory, or "memory aid," rather than "memory," is that the information has been "coded." In order to do Flintstone any good at all, he must remember his code, namely his mapping matrix. Moreover, his mapping matrix should be such that there is a one-to-one correspondence between two sets. Otherwise his retrieval of information will be partial at best. Just as he has to rely on an encoder matrix (or table), he will have to rely on a decoder to retrieve the message he has previously recorded, and as we have seen from our Theorem 1.1 the inverse mapping is guaranteed only when there is a one-to-one mapping. Figures 1.4.a and b illustrate a valid mapping, while c and d show useless coding.

Fingers		Number of Slashes		Fingers
Thumb	a	/	/	Thumb
Index	b	//	//	Index
Middle	c	///	///	Middle
Ring	d	////	////	Ring
Little	e	/////	/////	Little

1.4.a. Coder   1.4.b. Decoder

Thumb	/		
Index	/		
Middle	/	/	Thumb, Index, Middle, Ring, or Little
Ring	/		
Little	/		

1.4.c. Many-to-one Coder.   1.4.d. One-to-many Decoder.

Figure 1.4. (En)coders and decoders.

Communication

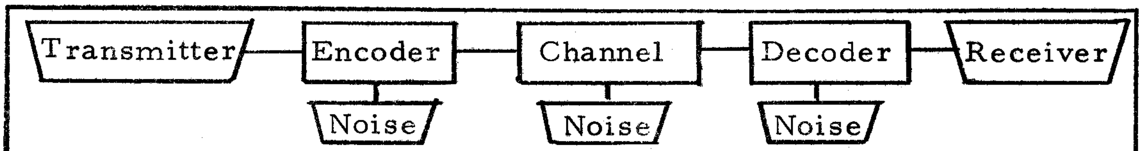
The markings on Flintstone's cave wall are just as observable by his friends as by himself. By teaching his friends the particular codes he is using, Flintstone can communicate with his friends. When this is done, the wall has become a "channel" for communication. Flintstone can leave a message on his wall and expect his friend to understand it when he sees the markings on the wall. The amount of information he can convey is limited by the three factors: how good his encoder is, how good his friend's decoder is, and how good is his channel. It is clear that some misunderstanding can occur in any of these three places: "I didn't mean it"; "I didn't say that" or "Didn't you see my note?"; or "You didn't understand me." These disturbances, called noise, are present even in the case of "external memory" but are usually corrected by the internal memory: "I don't remember writing this" or "I must have meant this."

By now it is obvious that the so-called model-building is an external memory device comparable to a communication channel. The amount of information that can be conveyed by the communication channel is never greater than that of information which the sender can retrieve from his own writing.

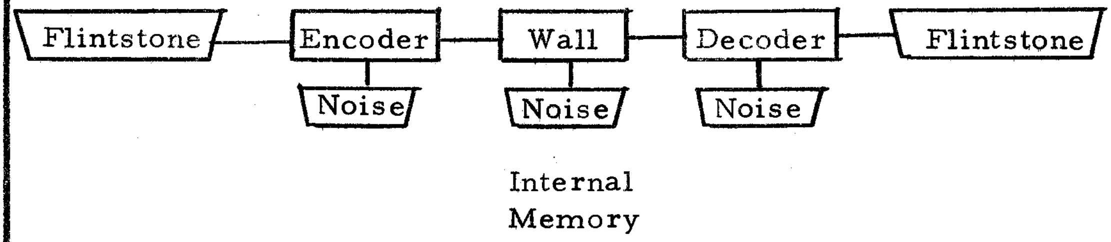
Threshold Logic Unit (TLU)

Figure 1.5.a shows the communication system between Flintstone and his friend. The comparison of this single-channel system to the case 1.5.b where the same coding is used for "external memory" will suggest at once how the effect of noise can be reduced. In 1.5.b we are really analyzing a two-channel situation (one external and one internal); the error introduced in one channel is corrected by the information received from the other.

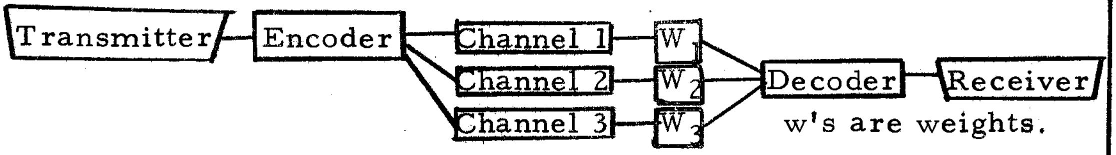
If Flintstone wants to be sure that his friend will interpret his message correctly, he can increase the number of channels and ask his friend to weigh the received signals to decode the message. This scheme is called a threshold logic unit and is illustrated in Figure 1.5.c. Let us next examine our own nervous system as a communication system with TLU's.



1.5.a. Single-channel communication system.



1.5.b. Flintston'es external memory.



1.5.c. Threshold Logic Unit.

Figure 1.5. Communication systems.

Neuron

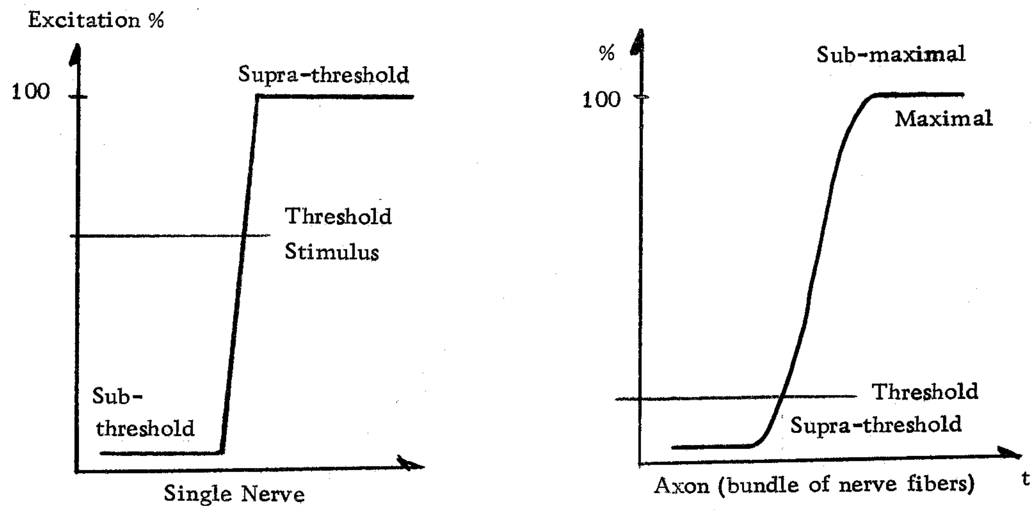


Figure 1.6.a. Threshold logic for nerve fibers.

Nerve Fibers TLU. Recent studies (Galambos, 1962; Arbib, 1964) indicate neurons to have threshold logic based on the accumulated responses from all the nerve fibers connected to the neuron through synapses. The nerve fibers conduct impulses as a binary signal (1 for -70mV difference between its inner (-) and outer (+) cells, 0 when the potential is collapsed by the flow of ions between the inner and outer cells) conducted at about 60 mph through the axon. The glia cells between nerve fibers seem to retain information sent by the nerves.

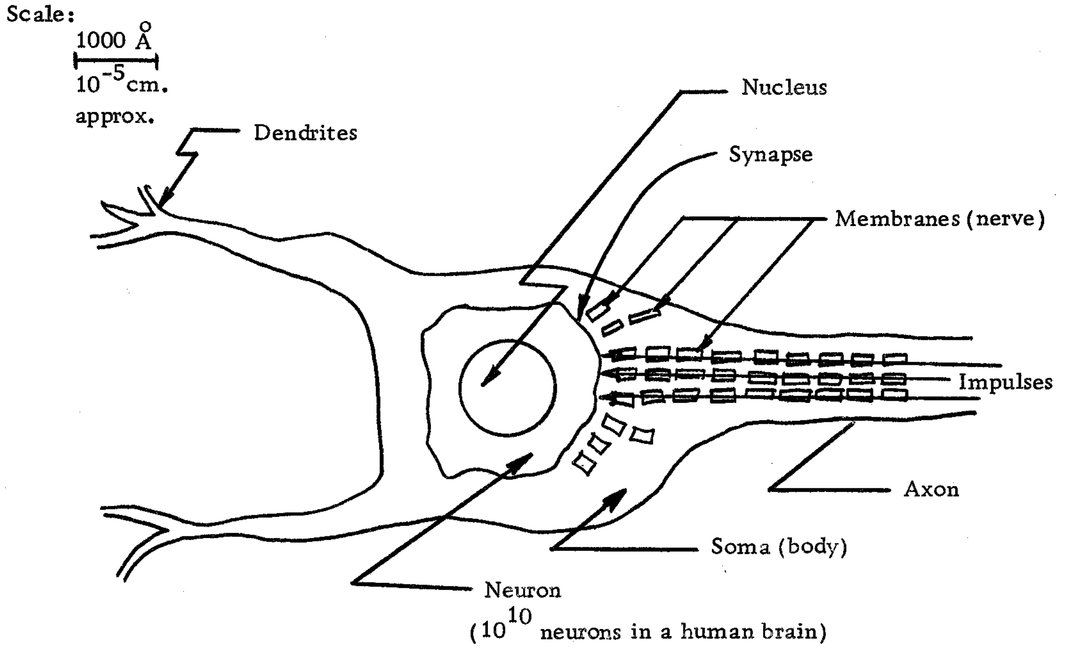


Figure 1.6.b. Schematic drawing of a neuron.

A neuron fires only if the total weight of the synapses which receive impulses in the period of latent summation exceeds the threshold (Arbib, 1964).

Membrane actions. Each membrane composing the nerve fiber appears to be at rest when there are approximately 70mV potential difference between the inner and outer cells. (Thus, state 1 is normal, and 0 when fired.) This difference in potential seems to be caused by the physical composition of the membrane which separates the inner and outer cells and which lets smaller ions pass through it but holds the larger ones on their original side. Thus, inside the membrane we have approximately $410K^+$, $49Na^+$, $40Cl^-$, and other ions of both polarities, while on the other side (outside the membrane) the composition of ions at rest is $22K^+$, $440Na^+$, $560Cl^-$ etc. When the nerve becomes active, it suddenly seems to abandon this property of holding larger ions (Na) from entering the cell. When Na^+ is free to enter the membrane, the unbalance of charge is destroyed and this message is conveyed along the nerve fiber until the synapse is reached. How this signal is transmitted to neuron (or muscle) is an unanswered question.

Neuron as an automaton. Let us quote what Wiener has to say about this action of neuron and nerve fibers (1948):

> It is a noteworthy fact that the human dna animal nervous systems, which are known to be capable of the work of a computation system, contain elements which are ideally suited to act as relays. These elements are so-called neurons or nerve cells. While they show rather complicated properties under the influence of electrical currents, in their ordinary physiological action they confirm very nearly to the "all-or-none" principle; that is, they are either at rest, or when they

"fire" they go through a series of changes almost independent of the nature and intensity of the stimulus. There is first an active phase, transmitted from one end to the other of the neuron with a definite velocity [(1 to 100 m/sec.)], to which there succeeds a refractory period during which the neuron is either incapable of being stimulated [(approx. 1 milli-sec.)], or at any rate is not capable of being stimulated by any normal, physiological process. At the end of this effective refractory period, the nerve remains inactive, but may be stimulated again into activity [(at about 20mV vs. 70mV negative inside when at rest)].

Thus the nerve may be taken to be a relay with essentially two states of activity: firing and repose. Leaving aside those neurons which accept their messages from free endings or sensory end organs [(some mechanical to electrical as in ear or abdomen, some chemical as in nose)], each neuron has its message fed into it by other neurons at points of contact known as synapses. For a given outgoing neuron, these vary in number from a very few to many hundred [(billions)]. It is the state of the incoming impulses at the various synapses, combined with the antecedent state of the outgoing neuron itself, which determines whether it will fire or not. If it is neither firing nor refractory, and the number of incoming synapses which "fire" within a certain very short fusion interval exceeds a certain threshold, then the neuron will fire after a known, fairly constant synaptic delay (Wiener, 1948).

This rather lengthy quote also points out the advances made by neurophysicists within recent years. The notes within [()] are from Dr. Robert Galambos book (1962)[1].

---

[1] In passing, we would like to note that Professor Galambos (psychology and physiology at Yale) interrupted his career to obtain his medical degree after he had received his Ph. D. from Harvard in 1941. He says "Medical training gives point and direction to the efforts of many biological investigators, and this it certainly has done for me." He received his M. D. from Rochester Medical School in 1946.

The descrete time model of neuron as an automaton is called McCulloch-Pitts Model in contrast to the continuous differential model by Peter H. Greene (1962). This concept of neuron as an automaton will be developed further in Chapter VI (page 200).

Transformation Systems

> Thus what usually happens is that the two systems, biological and model, are so related that a homomorphism of the one is isomorphic with a homomorphism of the other. (This relation is symmetric, so either may justifiably be said to be a "model" of the other.) (Ashby, 1963).
>
> The engineer feels he has modeled a system when he actually constructed an apparatus which he can hope will behave similarly to the original system. The mathematician, on the other hand, feels that he has modeled a system when he has "captured" some properties of the system in precise mathematical definitions and axioms in such a form that he can deduce further properties of this "formal" (i.e., mathematical) model; thus, hopefully, explaining known properties of the original system and predicting new properties (Arbib, 1964).

Group Theory

Definition 1.9. Empty set. An empty set is a set containing no element.

Definition 1.10. Identity element. In a set S with a binary operation $\boxtimes$, an element e is called an identity element for $\boxtimes$ if $e \boxtimes a = a$ and $a \boxtimes e = a$ for every element a in S.

Let us note that e does not have to be a numerical 1. For example, in the case of an addition, e is 0 rather than 1: a + o = a, and 0 + a = a (for an integer a). e is 1 for the multiplication: $a*1 = 1*a = a$. In Figure 1.2.b we had a as an identity element, whereas it was e in Figure 1.3. (e = 5).

Theorem 1.2. Uniqueness of an identity element. In set S with a binary operation $\boxtimes$, the identity element e for $\boxtimes$, if it exists, is unique.

Proof of Theorem 1.2. Suppose that the theorem is false, then there must be at least two identity elements e' and e'' for the binary operation $\boxtimes$. From Definition 1.10, we must have $e'\boxtimes a = a = e''\boxtimes a$ for every element a in S. Let e'' be a: $e'\boxtimes e'' = e'' = a\boxtimes e' = e''\boxtimes e'$. Similarly, by letting e' be a, we have: $e''\boxtimes e' = e' = a\boxtimes e'' = e'\boxtimes e''$. The comparison of the two equalities forces $e' = e''$ to be the only possible solution. Q.E.D.

Definition 1.11. Associative axiom. Let S be a set with a binary operation $\boxtimes$. The operation is said to be associative if, given any elements a, b, and c in S, we have:

$$(a\boxtimes b)\boxtimes c = a\boxtimes(b\boxtimes c).$$

Example 1.2. In Figure 1.2.b we would have: (a+b) = b and (a+b) + c = b+c = d. Also a + (b+c) = a+d = d. Thus, associative axiom holds for that example.

Definition 1.12. Inverse element. Let S be a binary operation

$\boxtimes$ and an identity element e for that operation. If a is an element of S and if there exists an element a' such that $a \boxtimes a' = a' \boxtimes a = 3$, then a' is called an inverse of a.

Theorem 1.3. Uniqueness of inverse. Let S be a set with binary operation $\boxtimes$ and an identity e for that operation, where $\boxtimes$ is associative. Then no element of S can have more than one inverse.

Proof of Theorem 1.3. Let a' and a'' both be the inverse of $a \in S$. Then, $a' \boxtimes a = a'' \boxtimes a = a \boxtimes a'' = e$. By associative axiom, on the other hand: $(a' \boxtimes a) \boxtimes a'' = a' \boxtimes (a \boxtimes a'') = e \boxtimes a'' = a' \boxtimes e$, thus $a'' = a'$. Q.E.D.

Theorem 1.4. Concellation. Let S be a set with an associative binary operation $\boxtimes$ and its identity element e. Let a, b, c, and c' be the four elements of S, and c' be the inverse of c. Then if $a \boxtimes c = b \boxtimes c$, then $a = b$.

Proof of Theorem 1.4. By Definition 1.2., we have $c \boxtimes c' = c' \boxtimes c = e$. $a \boxtimes c = b \boxtimes c$ becomes $(a \boxtimes c) \boxtimes c' = (b \boxtimes c) \boxtimes c' = a \boxtimes (e) = b \boxtimes (e)$, or $a = b$. Similarly, $c' \boxtimes (c \boxtimes a) = c' \boxtimes (c \boxtimes b) = e \boxtimes a = e \boxtimes b$, or $a = b$. Q.E.D.

Definition 1.13. A group. A set G with a binary operation $\boxtimes$ is called a group if:

(1) The operation $\boxtimes$ is associative.

(2) G contains an identity element for the binary operation; and every element in G has an inverse in G for the operation $\boxtimes$.

Example 1.3. Our example in Figure 1.2.b satisfied both conditions. (1) was already verified in the example following Definition 1.11. The identity element was recognized as a. The inverses are easily located in the matrix by spotting the identity element and recognizing its row and column. For example, $a^{-1} = a$; $b^{-1} = d$; $c^{-1} = c$; $d^{-1} = b$. Since there is one identity element in the body of the matrix for each row and each column (2) is satisfied.

We can repeat the same observation in Figure 1.3. The identity element is 5. Thus, $1^{-1} = 4$; $2^{-1} = 3$; $3^{-1} = 2$; $4^{-1} = 1$; $5^{-1} = 5$. Thus, $1 + 4 = 5$; $2 + 3 = 5$; $3 + 2 = 5$; $4 + 1 = 5$; and $5 + 5 = 5$. Notice that we adopted the notation a^{-1} in favor of a' to indicate the inverse of a. Also, we shall use the term "Group Operation" or sometimes the term transformation to indicate the binary operation of a group. The term group will be used to denote both the set of elements and the operation that satisfy the Definition 1.13. Before we proceed further, we shall illustrate another example of a group which results from permutation rather than rotation as our addition examples have been. This is shown below in Figure 1.8.

*	a	b	c	d	e	f
a	c	a	b	e	f	d
b	a	b	c	d	e	f
c	b	c	a	f	d	e
d	f	d	e	b	c	a
e	d	e	f	a	b	c
f	e	f	d	c	a	b

Figure 1.8. A permutation example.

The identity element in Figure 1.8 is obviously b; thus the inverses are: $a^{-1} = c$; $b^{-1} = b$; $c^{-1} = a$; $d^{-1} = d$; $e^{-1} = e$; and $f^{-1} = f$. Note that our concept of $1 = 1^{-1}$ as being a unique element with its own inverse equal to itself is not valid in this general group. Indeed an integer field is a very special group.

Homomorphism and Isomorphism

Now that we know what a group is, we are in a position to define homomorphism and isomorphism. The properties of a group have not yet been explored to any extent. Since the entire Tableau Method is based on the theory of group, it is imperative for us to study them in detail. However, we realize that this task may be undertaken more satisfactorily in Chapter III where the Tableau method will be introduced for the first time. Meanwhile the readers are asked to be satisfied with the definitions of the Isomorphism and Homomorphism and some explanations to make them plausible enough for the pending discussion of Engineering and Management Systems.

Definition 1.14. Homomorphism. Let $f{:}G \rightarrow G'$ be a mapping of one group G to another G', with group operations \$ and #, respectively. Then f is called a homomorphism if

$$f(a\$b) = f(a)\#f(b)$$

for all a, b in G.

Definition 1.15. Isomorphism. An homomorphism f is called an isomorphism of isomorphic groups G and G' if there exists one-to-one mapping from G to G' such that:

$$f(a\$b) = f(a)\#f(b)$$

for their respective group operations \$ and #.

It is to be noted that an homomorphism may be either one-to-many or many-to-one. Also an isomorphism is clearly a special case of a homomorphism. Let us now look at some practical examples.

Example 1.4. Homomorphic relations. Let us examine the relationship: $f(a\$b) = f(a)\#f(b)$. In essence it seems to say that the mapping f done to the operation a\$b in group G, is equivalent of first mapping the components a and b into G' and then operating on them with a corresponding function #. If a couple married in India is transported to the United States, it should be equivalent of transporting the individuals to this country first and then have them married.

The application of Isomorphism to the Boolean algebra (named after George Boole who wrote "Investigation of the Laws of Thought" in 1854) resulted in the much-celebrated DeMorgan's Theorem. Turning to Figures 1.2. c, d, e, and f, we notice that if we substitute the mapping function f by the inverse function yielding 1 for 0 and 0 for 1, AND in G will become NAND in G', and OR in G becomes NOR in G'. Thus,

$f\ (a\ \$\ b) = f\ (a)\ \#\ f\ (b)$ may be written as:

NOR = NOT (a OR b) = NOT (a) AND NOT (b) or

$\overline{a + b} = \overline{a} \cdot \overline{b}$ in a perhaps more familiar notation

Similarly

NAND = NOT(a AND b) = NOT(a) OR NOT (b) or

$$\overline{a \cdot b} = \overline{a} + \overline{b}$$

This tautology is named after the English Mathematician Augustus DeMorgan (1806-1871).

Model Building

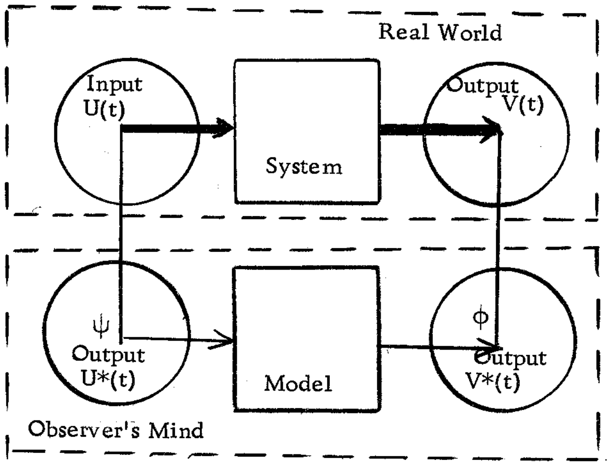
> The building of models for complex systems is unlikely to result in perfect isomorphism; in the case of exceedingly complex systems, that result will be by definition impossible to verify, if not to achieve. The system being studied has first to be simplified by a many-one transformation. This is what happened in Keynesian model of the economy studied by Tustin. That model is a homomorphism of the real system. Very often in operational research, and in cybernetics itself, we are studying an isomorphic machine which is not (as we would have liked) a transducer of the interaction of two large systems, but a model of that interaction studied through a homomorphism of each major system (Beer, 1964).

Experiments with black boxes. The concept of models has been extended to objects other than the members of the same species. In each case an object or a system of objects (actually, any object can be considered as a system in molecular level) has been considered as a black box and subjected to a variety of inputs. In other words, we try to "experiment" with the object as a "transducer" that converts our inputs into some outputs.

This approach, or "Black Box" concept is as basic as human nature itself, because of our need to use the physical environment as a media to explore the universe. When we have finally succeeded to construct a "model" of the system in question, we say that we have understood the system itself. This is precisely the same learning process that a newly born baby experiences.

A baby will touch a table, push it, bite it, and otherwise examine it until he can create in his mind a model of the table that will have the same set of attributes as his experiments afforded him: weight, taste, feel, etc. Since his mind is limited (both his memory and his power of association and analysis), he must confine himself to the set of attributes that are more obviously connected with the system and disregard all other sensations which cannot be attributed to the presence of the table. It is only after he has accumulated an adequate number of these models that he can classify them, generalize them, draw more abstracts from them, and finally be able to communicate with others that have similar models.

Figure 1.9 adapted from Iijima's article on pattern recognition (1966) illustrates the process by which a model is built in an observer's mind.



U and V are in the real space.
U* and V* are observation spaces.
The homomorphic transformations ψ, φ, mapping U into U* and V into V*.

Figure 1.9. Model-building with one input and one output variables (a simple transducer).

Models for communication. When communicating with another individual, the process must be reversed. The word "Table" must evoke in the correspondent's mind a similar model. "A table is sturdy" is the message from the sender conveying the outcome of an experiment performed on the "model" of the table in the transmitter's mind, such as tilting the table. The receiver can now use his model to try a simulation in his mind and compare the outcome. A result such as "I don't think so," is then conveyed back to the transmitter. This could result from the difference in their models, the difference in their simulations, or difference in their threshold value used as criteria.

Simulation and model are therefore very closely related. There are many authors such as: Hostein, Soukup, McMillan, and Gonzalez (1965) who openly admit that: "simulation occurs whenever a model of any sort is employed."

The term "simulation" is usually associated with a mathematical model, while an "experiment" is more frequently used with actual physical objects.

Transducers. When the system becomes complex, the homomorphism and eigenvalues are used to simplify the model. In a system "a house in earthquake" the model of a house composed of all the subsystems (transducers) is subjected to various inputs that may arise under the condition. The "characteristic" of the table as being "sturdy" may lead to the conclusion: "Hide under the table." Subconsciously the individual has classified subsystems having "sturdiness" as their characteristics and has selected the one which seemed to be the "optimal" shield. "A study of the real world thus becomes a study of transducers" (Goldman, 1953).

Modern Complex Systems

Engineering and Management

> 'I do believe,' said Alice at last, 'that they live in the same house! I wonder why I never thought of that before -- But I can't stay there long. I'll just call and say 'How d'ye do?' and ask them the way out of the wood... (Carroll, 1865).

Professional objectives. Engineers and managers share the same roof and are usually working for the same ultimate objective:

to maximize the overall profit and well-being of the particular organization. Both engineering and management may be defined as a profession dedicated to serve human society through the optimal physical transformation of limited resources. The fact that the human society is served indirectly through the particular organization for which they have been hired is only an incidental aspect of their professional beliefs.

What then is the distinction between the two professions? Perhaps the most basic difference is their interpretations of what is meant by limited resources and by optimal transformations.

To a traditional engineer, the limited resources denote anything that is convertible to another form of energy: mass, calory, chemical energies, nuclear energies, human energies, etc. The transformation means the change in form of energies, potential to kinetic, thermal to electric, electromagnetic to audio, etc. The optimality is measured by converting the input energy and output energy (or the desired part thereof) to common unit and finding their ratio. The transformation is considered to be more efficient as the ratio of useful output to the original input approaches unity.

$$\eta = \frac{\text{Useful Output}}{\text{Input}} \leq 1 \qquad \text{(Starr, 1964)}$$

To a manager, on the other hand, limited resources are anything that may become either assets or liabilities, in short anything that can be traded in for money. Their transformations are financial transactions which must produce higher values in outputs than in inputs:

$$\lambda = \frac{\$ \text{ value of the Output}}{\$ \text{ value of the Input}} \geq 1$$

Utility. The discrepency between the two criteria for the evaluation of performance is due to the difference in utility between the input and output. By a mere extraction of desirable parts from the mixture of desirable and undesirable (or a mixture of individually desirable parts, such as petroleum into gasoline and diesel), the utility of the output may rise appreciably. Unfortunately, "utility" is a highly non-linear (see discussion in Ch. VII) function about which very little is known to be practical. From the standpoint that managers recognized financial value as a closer evaluation of human wants and needs, we may concede that they have been more realistic than some traditional engineers.

Linearity. On the other hand, by the engineering assumption of linearity, especially in the form of Newtonian: Input = desired output + undesired output, together with the concept of homomorphic models, the engineers were able to create a true profession with a concrete body of knowledge.

Thanks to this linearity, without which neither Lagrange's equation, nor Kirchoff's laws would have been possible, engineers have gradually advanced their analytic abilities to where they can now handle systems of high complexities:

> Management has been sharply distinguished from engineering in literature even though both share common principles and philosophy. . . The engineer often sees the manager as superficial and lacking in intellectual rigor and depth, even though the manager must deal with systems of far greater complexity than those that the engineer designs. Conversely, the manager sees the technical man as narrow and lacking interest in people and social problems, even though the engineer is designing his technical systems on the basis of a body of philosophy, attitude, and theory that may help the future manager to better understand the complexity of social systems (Forrester, 1964).

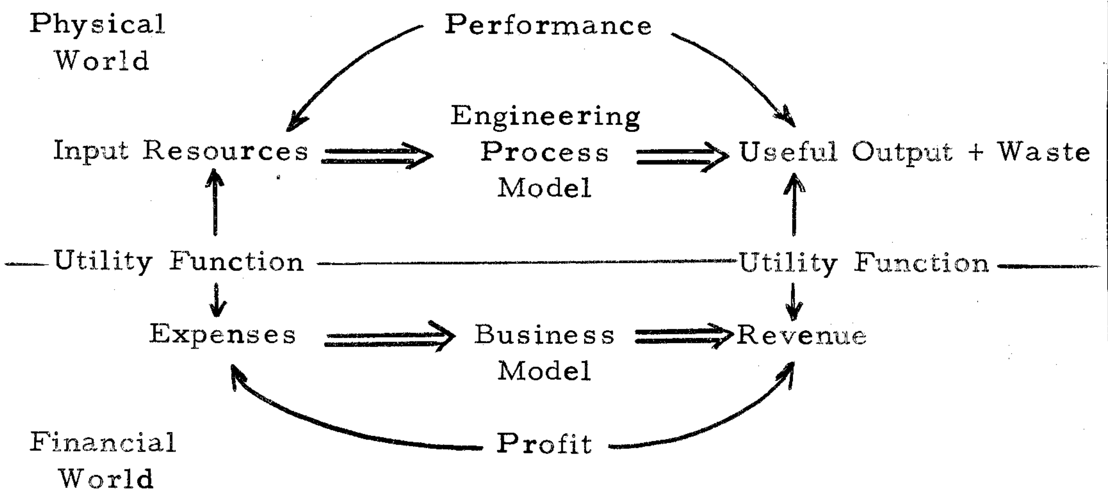


Figure 1.10. Engineering vs. business models.

What is common between an engineer and a manager is the concept of a "system" and its representation as a "model." An engineer communicates with another engineer using a "model," the same as managers would in their communications.

<u>The Ivory Towers of the Modern Babel</u>

> A major obstacle to adoption of the scientific approach seems to be a gap between management, who have the ability to define problems and to rate their relative importance, and the scientific analysts, who know the tools but may try to apply them without fully understanding the significance of the problem or without being able to communicate with the managers in order to find this out (Kozetsky and Kircher, 1956).

The traditional engineers and scientists who accepted linearity as a basic assumption in building their models have benefitted from the simplicity and ease of manipulations afforded by their linear models. They have become experts in linearly analyzing complex models. When a non-linear system had to be studied, they have simply looked for a linear substitute. Multiplications were handled by adding logarithms, combinatorial problems solved by adding or subtracting "bits" (1 bit = 1 binary digit = 1 binary combination. N combinations = $\log_2 N = 3.322 \log_{10} N$). Thus, for example, in information theory two independent classifications of 16 and 2 kinds each will give a total combination of $\log_2 16 + \log_2 2$ = 4 bits + 1 bit = 5 bits = 32 combinations. Similarly, differentiations and integrations were reduced to multiplications and divisions by the use of Fourier or Laplace Transforms, and when everything else failed they resorted to a piece-wise linear models.

However, it was not until managers realized how cleverly these "square-heads" were treating the problems of uncertainty by the use of probability theory and related statistical techniques that they accepted the benefit of adopting their techniques.

On the other hand, the increasing complexities and costs of their systems and related experiments have forced engineers to justify their work in terms of dollars and cents to the management for which they work.

Engineering models now had to be explained to managers in terms of the financial expense and resulting profits, while the management problems had to be explained in terms of mathematical models to the engineers and scientists. Unfortunately, this has not proved it to be an easy task. Even vocabulary used by the two had different meanings to the two professions. For example, "variance" for an engineer will be a statistical measure of central tendencies, while to a manager it means a simple difference between the budgeted and actual spending. When a manager talks about conducting an "Acid Test," he is not talking about HCl or H_2SO_4. Rather, he will want to take a numerical ratio of quick assets (cash, marketable securities, and account receivables) to current liabilities.

In the complex society of today, one must understand a complete set of jargons before being admitted into a profession, and a "profession" that has no profound speciality creates its own jargon to safeguard the possibility of being overtaken by a more powerful profession.

> These specialized fields are continually growing and invading new territory. The result is like what occurred when the Oregon country was being invaded simultaneously by the United State settlers, the British, the Mexicans, and the Russians -- an inextricable tangle of exploration, nomenclature, and laws (Wiener, 1961).

Nature and Management

> Let both sides seek to invoke the wonders of science instead of its terrors. Together let us explore the stars, conquer the deserts, eradicate diseases, tap the ocean depths and encourage the arts and commerce (Kennedy, 1961).

When the managers and engineers have finally succeeded in communicating with each other, their problem-solving task begins. The system they have elected should perform in an optimal manner under the given circumstances or environment. The distinction between the system and its environment is made on the basis of the objective for which the problem must be solved. The environment is simply "the set of all objects, a change in whose attributes affects the system, and also of those objects whose attributes are changed by the behavior of the system" (Hall, 1962). By considering the environment as a willful expression of Nature, we may consider the management problem as a Two-person game played against Nature. Von Neumann is considered to be the pioneer in the application of Game Theory to Economic problems (Von Neumann and Morgenstern,

1964) while Wald is credited for recognizing the decision problem as a disguised Two-person game against Nature (Wald, 1950). We hope the readers will not be offended by our use of the word Nature to indicate the environment imposing constraints on Management actions.

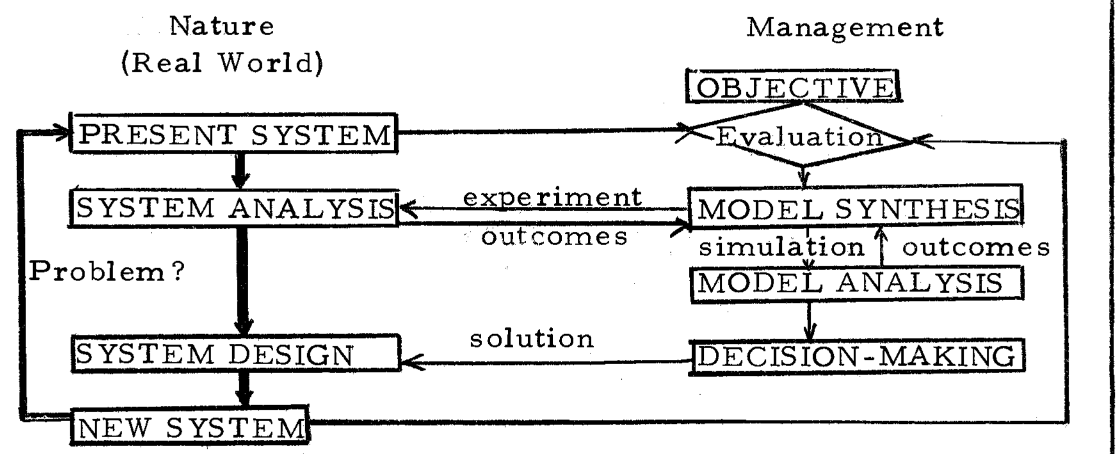


Figure 1.11. Problem-solving: simplified.

Figure 1.11 illustrates the simplest outline of a problem-solving procedure. Once the model has been built from the actual system, all the study is made on the model of the problem[2] until the decision is reached as to which of the alternatives suggested by the model simulation should be used, or equivalently, what should be the values of the parameters in the actual system.

The final product is a system which is a compromise between the design generated by management and the circumstances decided by the Nature. Greeniewski (1965) has termed this: "the subject of

---

[2] Also called hypothesis by many authors such as McMillan (1965), Bowman and Fetter (1961), etc.

planning."

A portion of Figure 1.11 is duplicated below in Figure 1.12 to illustrate this view.

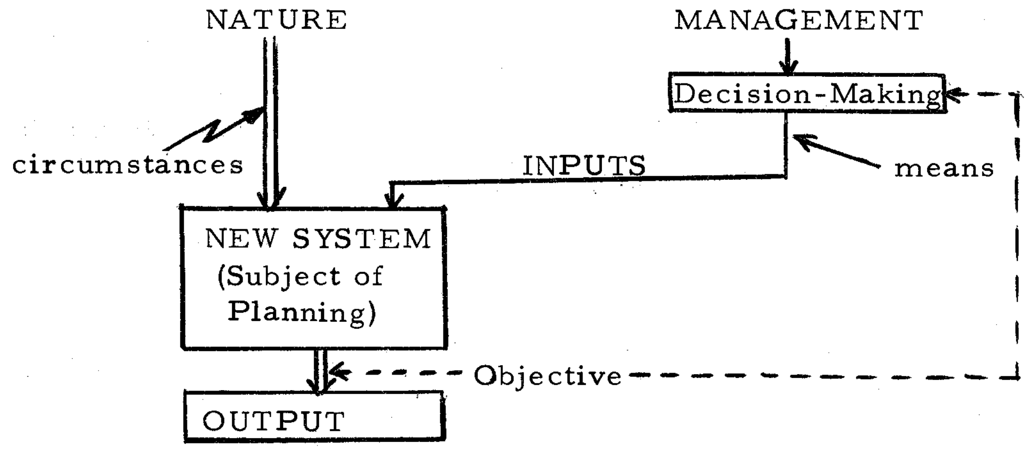


Figure 1.12. Greniewski's model for system planning.

Nature and Management respectively determine the uncontrollable and controllable portions of the inputs. Usually the Nature provides the materials (or energy) and time restrictions, while the management will determine the actual values of the parameters to be used in the system.

Systems Science and Cybernetics

What's in a name? That which we call a rose
By any other name would smell as sweet
(William Shakespeare: Romeo and Juliet).

At this point we would like to acknowledge the numerous attempts made by people in various professions to reconcile their differences and bring about a mutual understanding of what systems science should be.

Cybernetics

Nobert Wiener founded Cybernetics to encompass: "the entire field of control and communication theory, whether in the machine or in the animal. . . " (Wiener, 1961, p. 13). The term "Cybernetics" was coined from the Greek word "κυβερνήτης" which means "steersman" and from which the Latin word "gubernator,"[3] the French word "Gouvernail,"[4] and the English term "governor" were derived (Beer, 1964; Guilbaud, 1959).

> Wiener and his associates decided on this name for cybernetics in 1947, and he records that the "steering engines of a ship are indeed one of the earliest and best developed forms of feedback mechanisms." Plato had used the word cybernetics in his time, and Ampère[5] had borrowed the term also as a name for the science of government; but Wiener must take the final responsibility for the currency of this ugly word, and also the credit for its great aptness (Beer, 1964).

Systems Engineering

Many electrical engineers felt that their particular specializations have become too complex and comprehensive systemwise and

---

[3] Helmsman.

[4] A rudder in French.

[5] The French word for cybernetics is "Le Cybernétique."

that they should be given special titles. They became Control Systems Engineers, Communication Systems Engineers, Guidance Systems Engineers, Computer Systems Engineers, Information Systems Engineers, or just plain Systems Engineers.

Operations Research

The organization of the first Operations Research team is attributed to the Nobel Laureate Professor P. M. S. Blackett of the University of Manchester. This multi-discipline team organized in 1939 to help the British Army's Operational Research group was called "Blackett's circus." It is told by Dean Roy of the Johns Hopkins University (Flagle *et al.*, 1960) as having included "three physiologists, two mathematical physicists, one astrophysicist, one Army officer, one surveyor, one general physicist, and two mathematicians."

During the last War, a similar organization was established in each branch of the United States Armed Forces: "the Operations Analysis Group (OAG) with the Air Force, the Operations Evaluation Group (OEG) with the Navy, and the Operations Research Office (ORO) with the Army" (Flagle *et al.*, 1960).

The distinction between the "systems engineers" and the "operations researchers" are more historical than real. Dean Roy distinguishes the former as being "electronic-communicationsservomechanisms-human engineering-design" minded while the latter is more inclined toward "mathematical models, stochastic processes, statistics, probability, economics, and behavior science" orientation (Flagle et al., 1960).

Industrial Engineering

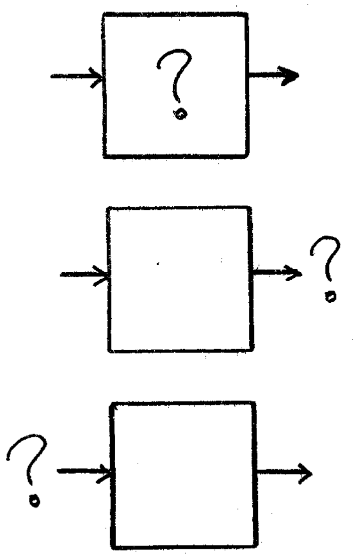
To an industrial engineer, finally, all these claims seem somewhat of an imposture upon their definition of industrial engineering:

> Industrial Engineering is concerned with the design, improvement, and installation of integrated systems of men, materials and equipment. It draws upon specialized knowledges and skill in the mathematical, physical, and social sciences together with the principles and methods of engineering analysis and design, to specify, predict, and evaluate the results to be obtained from such systems (American Institute of Industrial Engineers Creed).

It is, in fact, based on this Creed that this treatise is presented justifiably under the auspices of Industrial Engineering. However, we have, for the purpose of convenience and in order to avoid any possible misunderstanding, chosen the term "Systems Theory" to describe the composite of all knowledges and techniques dealing with "Systems."

Furthermore, we shall make a blunt assumption that Cybernetics will be primarily related to the Model Synthesis, Operations Research with Decision-Making, and Systems Engineering with the

System Design phase of "Systems Theory." Our postulate is that the Tableau Method may serve as a common language to link these three phases of Systems Theory.



Model Synthesis:
- Tableau Method:
  - Matrix Representations
  - Network Representations

Model Analysis:
- Tableau Manipulations
- Eigenvalue Problems
- Transforms
- Linear Models

Decision-Making:
- Dynamic Programming
- Linear Programming
- Utility Theory

Figure 1.13. Systems Tableau techniques.

Our first attempt will be to define what we should mean by the term "System."

II. SYSTEMS

Our little systems have their day;
They have their day and cease to be:
They are but broken lights of thee.
And thou, O Lord, art more than they.
(From "In Memoriam" by Lord Alfred Tennyson, 1869).

The Black Box Concept

A system is most commonly represented by a black box with an input and an output. Too often it has been used as a convenient vindication to conceal one's ignorance about a mechanism too complex to analyze. We also see it being used as an "action" word needed in an attempt to upgrade a concept, especially when the concept is so poor that it does not warrant any other means of salvation. Probably there are very few terms, if any, used under meanings and implications as varied as this word "system."

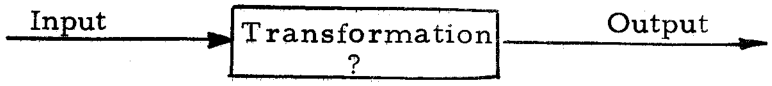


Figure 2.1. A black box concept

However, since the development of servomechanism, control theory, automation, cybernetics, management sciences, systems engineering, and data processing, the term "system" has slowly come to bear an identity distinctly its own in this family of systems theory. Several definitions advocated by persons actively engaged in systems theory will be examined in order to identify these distinct features.

The Definitions of a System

The most general and the least sophisticated definition of a system is likely to be the one afforded by the Webster's dictionary: "A number of things adjusted as a connected whole; a scheme, plan, or method."

W. D. Rowe, the past chairman of the Systems Science and Cybernetics Group of the American Institute of Electrical Engineers (prior to becoming IEEE) has a similar definition: "A system is any large collection of interacting functional units that together achieve a defined purpose" (Rowe, 1965).

The concept of "interacting functional units" may be explained in terms of objects (components of the system) and their attributes (properties of the objects). Arthur D. Hall, the new editor of the Systems Science and Cybernetics Group of the Institute of Electrical and Electronics Engineers (IEEE) defines a system as: "A set of objects with relationships between their attributes" (Hall, 1965).

J. S. Frame and H. E. Koenig take a further step by defining discretely the components and their attributes:

> By definition, a system is a collection of discrete components, each having certain definable characteristics, together with a prescribed pattern of interconnections

> or interrelations...
>
> In any case, the system can be represented by a diagram as that shown [(in Figure 2.2)][6] where a point of contact between any two regions represents a junction or interface between two components, and is referred to as a terminal of the components (Frame and Koenig, 1964).

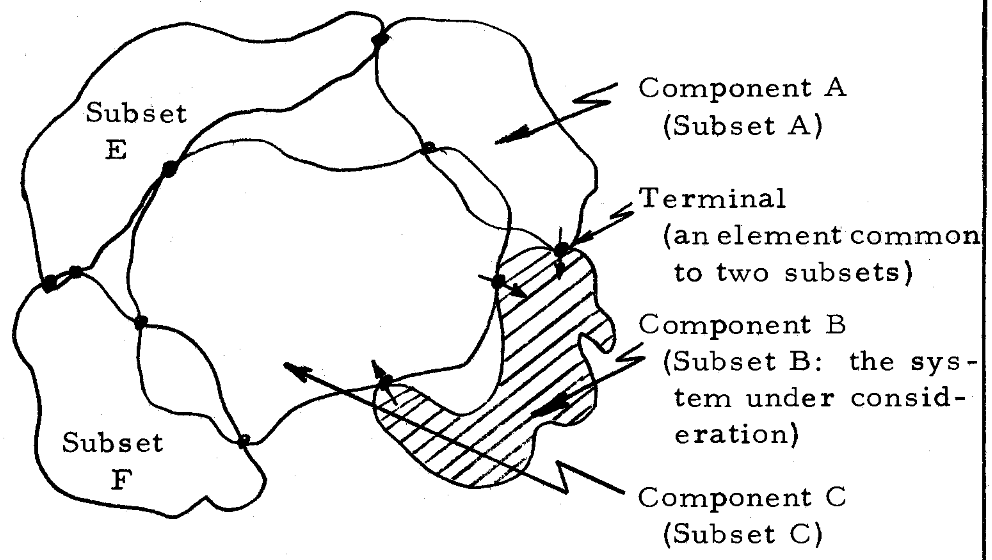


Figure 2.2. A system of interconnected components.

Once the definition of a system is recognized as an abstraction based on the attributes of objects, it becomes possible to define a system in terms of its model.

> A system is a mathematical abstraction that is devised to serve as a model for a dynamic phenomenon. It represents the dynamic phenomenon in terms of mathematical relations among three sets of variables known as the input, the output, and the state (Freeman, 1965).

---

[6]Brackets and parentheses within a quote [( )] will be used to indicate remarks made by the author and not by the person being quoted.

We notice that Freeman's definition is almost identical to our definition of an automaton in Def. 1.8. It also reminds us of Wiener's concept of today's automata, namely his Cybernetics:

> ...we deal with automata effectively coupled to the external world, not merely by their energy flow, their metabolism, but also by a flow of impressions, of incoming messages, and of the outgoing messages. The organs by which impressions are received are the equivalents of the human and animal sense organs (Wiener, 1948).

The system component B from Figure 2.2 may be redrawn in the form of a black box as shown in Figure 2.3.

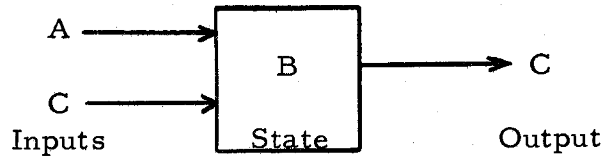


Figure 2.3. Black box corresponding to Figure 2.2.

At this point, it may be interesting to recall the definition of a graph and compare it to those we have found for a system.

> There are three items which characterize a graph. First, there is a set X of elements called points or vertices: $X = \{x_1, x_2, x_3, \ldots x_n\}$; second, there is a function Γ mapping X into itself; third, there is a set U of arcs joining the elements of X according to the rule Γ.
>
> Any two of these items are sufficient to completely define a graph. Therefore, a graph is usually expressed as: $G(U, \Gamma)$ or $G(X, U)$ (Inoue, 1964).

If X were to be the set of attributes, U, the set of relations, and Γ, the transformation, this definition of a graph would be essentially identical to that of a system. Moreover, the two common expressions of a graph: $G(U, \Gamma)$ and $G(X, U)$ correspond to our concepts of systems analysis and design!

Therefore, it seems logical that any system that can be defined by an input set, an output set, and the relationships between them ought to be presentable in a graphic form.

In the example used in Figures 2.2 and 2.3, we have: $x = \{A, B, C\}$; $U = \{(A, B), (C, B), (B, C)\}$. The transformation rule Γ has not been defined yet, but obviously corresponds to the concept of a mapping matrix as we have seen in Chapter I.

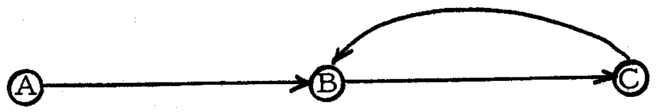


Figure 2.4. A graph representation.

One of the most sophisticated descriptions of a system is given below by S. S. Sengupta and R. L. Ackoff of the University of Pennsylvania. This particular definition recognizes "decision-making" as an inherent function of a system.

> Insofar as the purposeful entity is a <u>system</u>, it must contain at least two interdependent parts, each of which has a different function to perform and has an objective associated with its function. Furthermore, communication between these parts must be possible. Therefore, we conceive of a system as a set of activities that are connected both in time and space by a set of decision-making and behavior evaluation practices (Sengupta and Ackoff, 1965).

The recognition of time and space relationships between systems components, the acknowledgement of feedback control, and the identification of communication (rather than material flow) as the essential links make this description one of the most elegant definitions of a complete system. The recognition of information communication is particularly important. A signal or information may be copied and be dispatched to several receivers at once even though it is true that no transmission of information can occur without some form of energy. A process or a material flow, on the other hand, can only be used by one person at a time. The energy will have to be split among users. A product must undergo one operation at a time. Each object is unique and cannot be exactly duplicated. On the other hand, a piece of instruction copied from an original is as good as the original as long as the error (i.e. noise) does not alter the meaning.

Essential Properties of a System

Complexity

The analysis of the common traits in the above definitions allows us to make plausible generalizations. The first common property of a "system" lies in its complexities. Not only each element composing the system should act as a "transducer" thus relating two other elements, but the system itself should serve as a subsystem (another transducer) for a larger system. The need for an integrated systems theory arises precisely from the inability of older techniques to handle complex systems:

> In the study of some systems, however, the complexity could not be wholly evaded. The cerebral cortex of the free-living organism, the ant-hill as a functioning society, and the human economic system were outstanding both in their practical importance and in their intractability by the older methods. So today we see psychoses untreated, societies declining, and economic systems faltering, the scientist being able to do little more than to appreciate the full complexity of the subject he is studying. But science today is also taking the first steps towards studying 'complexity' as a subject in its own right (Ashby, 1956, p. 5).

Observability and Controllability

Because of the black box concept as the basis for understanding a system, it is evident that a system must respond at least partially to the experiments conducted on them. In other words, its inputs must be at least partially controllable while its outputs must be at least partially observable. A system that satisfies this condition will be called an (partially) open system.

> Most. . . systems are open, meaning they exchange energy with their environments. A system is closed if there is no import or export of information, heat or physical materials, and therefore no change of components (Hall, 1962, p. 69).

From these considerations, it becomes difficult for us to accept the statement that Ashby makes about Cybernetics (1963, p. 3).

> ...whether the system is closed to energy or open is often irrelevant; what is important is the extent to which the system is subject to determining and controlling factors. So no information or signal or determining factor may pass from part to part without its being recorded as a significant event. Cybernetics might in fact, be defined as <u>the study of systems that are open to energy and closed to information and control--systems that are 'information-tight.</u>'

The first part of his statement clearly shows that his system is meant to be controllable and observable. The information must, therefore, enter the system and leave the system, without being affected by the energy considerations. This is precisely the problem that electrical engineers have been tackling for years. They were concerned about an electrical system that would accept information, process it and return it without the physical restrictions imposed by energy consideration. A transistor radio is practically a closed system energy-wise. The amount of energy received from the outside is almost negligible, and so is the mechanical energy dissipated by an earphone. The information received, on the other hand, has undergone a drastic transformation from a controllable input (electromagnetic wave) to observable output (audio signal). To build a circuit that will have an infinite input impedance and essentially null output impedance has been the dream of electronic engineers. Such a circuit would be able to accept a faintest signal and distribute its information to as many users as desired without having it distorted due to the limited energy consideration.

The only time a closed system would be of interest to us is when we are a part of the system itself. But this is possible only after we have studied each component as a transducer, an obviously open system.

Possibly what Ashby has done was to misunderstand the discussion led by Wiener in recognizing the relationship between Cybernetics and what we call "electronics" (and what is called "weak currents" electrical engineering in Germany and Japan), as opposed to the "strong current" power engineering. The power engineering deals with the study of conservative systems (that is closed systems in which mass, energy, and momentum, etc. are conserved). In electronics the conservation of energy is secondary to the processing of information. (Energy-wise, a computer is a very inefficient system if all we are interested in is to have characters printed on a sheet of paper.) But this does not make a system closed informationwise. A piece of information is an abstract quantity. To admit that a system is closed is to admit that what is called information is really not "meaning" but energy carrying the information. This is because "meaning" can be distributed to a multitude of receivers without losing its content. Energy would have to be infinite if each receiver is to receive as much as the original energy carrying the input information.

Of course, this does not subtract anything from the practical usefulness of a "closed system" as a concept in systems analysis and design. We would only wish to have it correctly interpreted.

The fact that a system cannot be observed without it being perturbed may be true from a Quantum Mechanical point of view but certainly not from the Cybernetics point of view, as portrayed by Wiener. What he wanted to point out was the importance of the "closed loop" or feedback theory and information theory as tools for this new science. The servomechanism theory of a closed loop would have no use in a closed system. Wiener is also credited as the first to bring in uncertainty as an intrinsic characteristic of a complexed system.

> This something extra he identifies as uncertainty, not only the uncertainty of Heisenberg[7], but more importantly the uncertainty of statistics, which he relates back to the statistical nature of information theory and of the inputs to control systems (Goode and Machol, 1957, p. 393).

Again, it is obvious that Cybernetics assumed its systems to be open.

We conclude the discussion of Observability and Controllability by illustrating their concepts as shown by Freeman (1965).

---

[7]See page 80 of Chapter III for the statement of the principle of uncertainty.

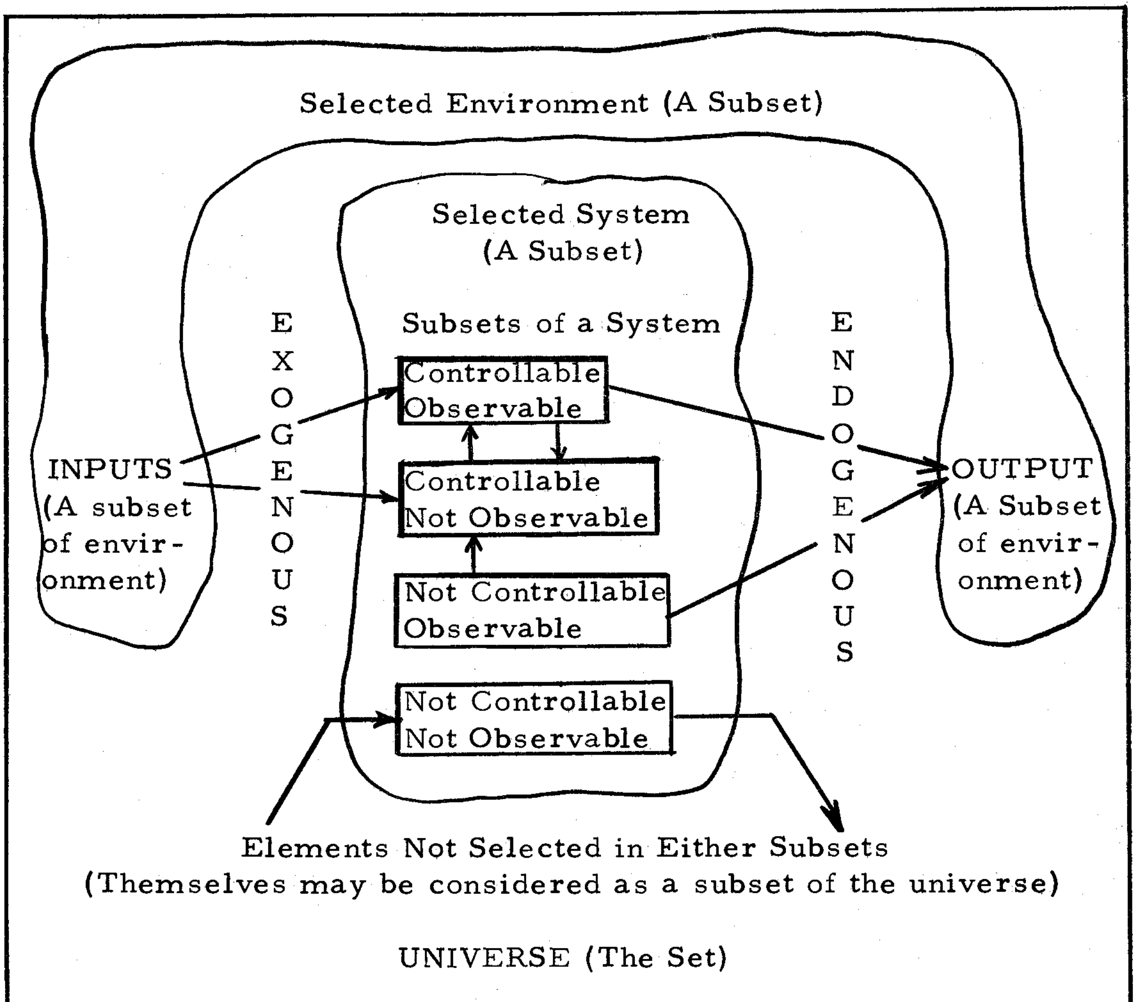


Figure 2. 5. A system and its environment.

Endogenous output is an element in the environment dependent on (mapped on by) the selected system. Exogenous input, by contrast, is an independent variable of the system.

These terms and others in the above figure will be defined later. Meanwhile, this figure may be taken as their intuitive descriptions.

Time Dependence

> "...Everything changes but change itself" (Kennedy, 1963).

There are several reasons why Time plays such an important role in Systems. First, the relationships between the components of a system are described in some logical manner, technological if not mathematical ordering. This ordering concept is possible only because we assume time to be a uni-dimensional quantity: "Within any world with which we can communicate, the direction of time is uniform" (Wiener, 1961). Thus an open system must be time-dependent.

Another reason for the need of the system to be time-dependent is that the system must be physically realizable. All physically realizable systems must be non-anticipatory. Since a system cannot be controllable and observable if it is not physically realizable, we may consider this to be a corollary of the first argument.

For example, it is possible in designing an electrical network in its frequency domain, to arrive at a design that requires physically unobtainable components when the results have been transformed back into time domain. This may be manifested by a requirement that negative inductance or capacitance is required. A further investigation may reveal that the designer, in such cases, was trying to synthesize a system that depended on future information.

To restate the requirement, a system is said to be nonanticipatory (or physically realizable), if its state (x) and output (y) at any time (t_o) may be a function of only those input values that have occurred at time $t < t_o$. Thus, a nonanticipatory system cannot respond to input values until after their occurrence.

> A basic characteristic of any 'dynamic phenomenon' is that the behavior at any time is traceable not only to the presently applied forces but also to those applied in the past. The state of the system represents the instantaneous content of the 'memory' of the system in which the effect of past applied forces is stored. The output is determined by the state and presently applied input (Freeman, 1965).

Ashby, on the other hand, violently opposes the concept of a "memory" as an intrinsic property of a system. "The possession of 'memory' is not a wholly objective property of a system" (Ashby, 1963, p. 116).

We would also like to call attention to the close similarity presented by Freeman's model of a system and our modern digital computer configuration. The comparison makes it clear why an electronic computer is such a powerful tool for systems analysis. Not only is the computer system itself a system in its own right, it has the capability of subdividing its own memory to simulate subsystems acting as components of a larger system. It also meets the criteria of complexity, controllability, and observability.

Before leaving this topic of time, we should mention the well-known paradox of "Maxwell demon" discussed by Wiener (1964, p. 57).

> Let us suppose a gas in which the particles are moving around with the distribution of velocities in statistical equilibrium for a given temperature. For a perfect gas, this is the Maxwell distribution. Let this gas be contained in a rigid container with a wall across it, containing an opening spanned by a small gate, operated by a gatekeeper, either an anthropomorphic demon or a minute mechanism. When a particle of more than average velocity approaches the gate from compartment A or a particle of less than average velocity approaches the gate from compartment B, the gatekeeper opens the gate, and the particle passes through; but when a particle of less than average velocity approaches from compartment A or a particle of greater than average velocity approaches from compartment B, the gate is closed. In this way, the concentration of particles of high velocity is increased in compartment B and is decreased in compartment A. This produces an apparent decrease in entropy; so that if the two compartments are now connected by a heat engine, we seem to obtain a perpetual motion machine of the second kind. ... (in solving the paradox) the demon can only act on information received, and this information... represents a negative entropy... under the quantum mechanics, it is impossible to obtain any information giving the position or the momentum of the particle, much less the two together, without a positive effect on the energy of the particle examined. (See Heisenberg's principle on p. 80 of Chapter III)

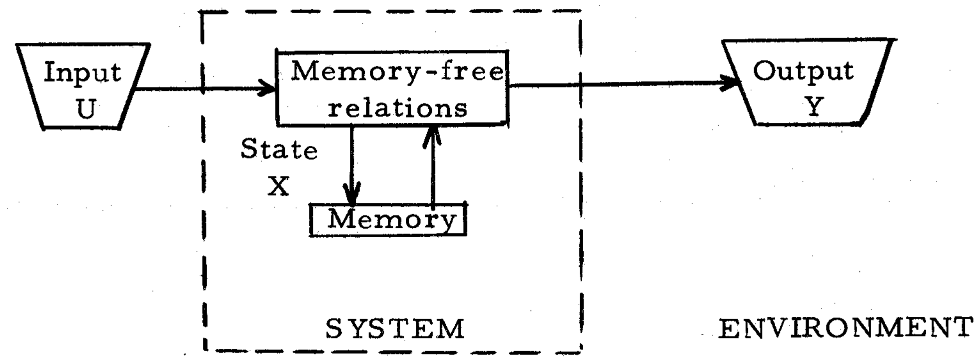


Figure 2.6. Freeman's representation of a system. U, X, and Y are vector quantities.

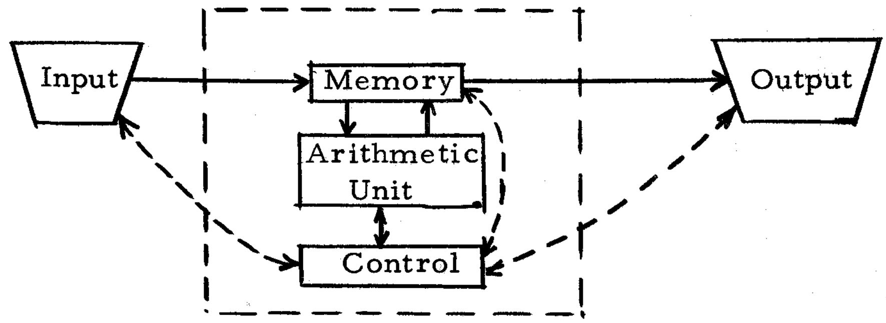


Figure 2.7. A model of an electronic digital computer.

Suboptimization in a System

The last and probably the most important feature of a system is the existence of a purpose for which the system was conceived. The degree of attainment of this objective can be measured and compared with "criteria" to evaluate the effectiveness of the system. This implies, in turn, that a system can be optimized by evaluating the magnitude of input necessary to attain a common effectiveness with respect to the objectives.

Since a system is an interacting unit, it may also become a sub-system of a larger system with a different objective. The larger system must have a "memory" that will be longer than that of the subsystem (or at least as long in retention as the longest of subsystem memories), and its objective is also likely to be of "longer" range.

Each subsystem will strive to optimize its own objective within the constraints imposed on them. The over-all system will try to maximize its over-all objective. This problem of suboptimization is averted by making a part of its inputs uncontrollable by the subsystems and to be controlled directly by the input into the over-all system.

These quantities may be illustrated by the example of the free-enterprise system in which we live. Each industry or organization is a subsystem created to attain its own objective, usually to bring financial profit to its owners. The overall effect, on the other hand, is controlled by the government to produce most benefit to all the citizens.

Components of a System

Elements

The elements of a system may be divided into "variables" and "parameters." Variables are the characteristic quantities that may assume different values during the experiments. For each transducer we shall have a set of independent variables and a set of dependent variables. For the overall system, the independent variables are referred to as "exogenous" or input variables, while the dependent variables are termed "endogenous" or output variables (Figure 2.5).

The quantities which stay constant during the simulation or experiment will be called "parameters." The distinction between a parameter and an independent variable is more theoretical than practical. It depends largely on how one defines an "experiment." $F = m\ddot{x}$ may be simulated on an analog computer by specifying the parameter m. If the experiment is to find the proper value of m to yield the desired F, $\ddot{x}$ may be chosen as a parameter and m(t) varied as the experiment progresses.

The distinction between an endogenous and an exogenous flow is equally subject to controversies. If the injection of electrons is considered to be an input, would the injection of holes (lack of electrons) be considered an input or an output? Authors such as Dantzig prefer to use the term "exogenous flow" to express both inputs and outputs (Dantzig, 1963, p. 45).

Transfer Functions

The relationship between the independent and dependent variables of a transducer is usually expressed as a transfer function, sometimes called "transmittance."

$$\text{Transmittance} = \frac{\text{Output Function}}{\text{Input Function}}$$

A numerical value that can characterize this function is akin to the eigenvalue for a given system. Often a linear expression will be found in the frequency domain that will describe this transfer function and is referred to as a characteristic function.

Fan-in and Fan-out

The outputs from several transducers may become inputs to several other transducers. This many-to-many relationship could be considered as a transducer in its own right. However, it is usually more convenient to distinguish this type of transformations from the conventional transducers. The reason is that by selecting transducers as the subsystems of the larger system we have given up our right to interfere with the internal operation of transducers except through their inputs. We can exercise our control in the system design at the junction of outputs to inputs.

The fan-in decisions are usually simple relations such as: AND (all outputs from the preceding transducers must be in before fan-out decisions can be made); OR (any input received into the junction from a transducer is good enough for deciding a fan-out); or NOR (not one or the other).

The fan-out decisions are more elaborate in general: MAX (choose the largest of all outputs from the preceding transducers and use it as an input to the succeeding transducers): MAXIMIN (choose the output vector from the preceding transducers that has the largest of the smallest component): AND (use all outputs from the preceding transducers), etc.

The readers are cautioned against including the decisions that involve the "anticipatory" knowledge of the following transducers. For example, we cannot say "feed the input to the most promising transducer." In such a case, what we are really deciding is: "if the output from the experiment conducted on a model is higher than a particular threshold level, let the preceding transducer be connected to the particular transducer." Though the distinction may seem nebulous, we hope the importance of "non-anticipation" will become clearer as we proceed with more rigorous models.

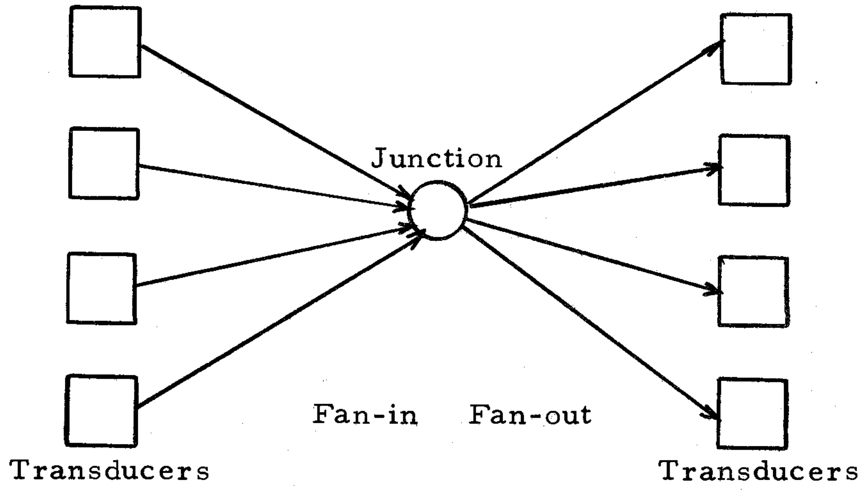


Figure 2.8. Fan-in and fan-out.

The Proposed Definition of a System

Let us now summarize our discussion into a proposed definition of a system:

> A system is a purposeful, dynamic entity whose inputs are at least partly controllable and whose outputs are at least partly observable; it is composed of interacting elements, each of which possesses the characteristics of a complete system.

We shall use this definition in Chapter III to formulate our Tableau Method. We have deliberately withheld any concrete example or too involved mathematical discussion in the first two chapters to allow a systematic introduction of our proposed approach. As it has been made clear in our previous discussions, most useful and rigorously developed techniques are those concerned with linear systems. The fields that may prove to be most fruitful in the future, on the other hand, are those that are not strictly bound by the linearity conditions. In order to keep our Tableau method as versatile as possible, we shall try to avoid any such particular restriction to be imposed on the formulation of Tableau except for those explicitly stated in the above definition. In the chapters following, we shall try to substantiate qualitative statements and observations made in the first two chapters by more quantitative and concrete examples and, at the same time, by making use of the Tableau Method whenever applicable.

III. FORMULATION OF TABLEAU METHOD

And there shall come forth a rod out of
the stem of Jesse, and a Branch shall grow
out of his roots. . . (Isaiah 11:1).

Basic Philosophy

Systems Theoretical Considerations

In Chapters I and II, we have reviewed some of the basic concepts that are at the basis of today's systems theory. In Chapter I, we have borrowed from pure and natural sciences some basic definitions and theorems that are at the root of any branch of systems theory. On the other hand, we have been very careful not to accept any statement, definition, or theorem from Chapter II where contemporarily developing theories were discussed.

The domain of disciplines represented by today's systems theorists is diversified indeed. There were first the Industrial Engineers, then various types of Electrical and Electronic Engineers, Military, Statisticians and Operations Researchers, Computer Designers and Programmers, Systems and Management Analysts, Medical Doctors and Bio-Engineers, Cyberneticians (if such a term exists), Social Scientists, Economists, more engineers and others (including medical doctors and journalists), and presently, we are facing a sudden inflow of Chemists wishing to enter this field (Leondes, 1966).

It is needless to say that each of these groups have brought with them a new viewpoint and an accompanying language to explain it. The result is not unlike Japan of the first Centuries when Chinese Budhist missionaries crossed the Japan Sea to preach their Faith. They taught prayers translated into Japanese using their Chinese prayer books. (Japanese 2000 years ago was a well established conversational language but the written language did not become popular until the arrival of Chinese missionaries.) The poor converts had to learn the Chinese characters with both Chinese and Japanese pronunciations, a fact that has resulted in such long-lasting confusion that puzzles foreign visitors even today.

Inspite of the difficulties, we have managed to construct a definition for what we understand by the term "system." Let us now analyze the definition stated in Chapter II.

> <u>Definition 3.1. A system.</u> A system is a purposeful, dynamic entity whose inputs are at least partly controllable and whose outputs are at least partly observable; it is composed of interacting elements, each of which possesses the characteristics of a complete system.

This definition is rather remarkable from several points of view as we shall see from our discussion in the next section. Meanwhile, let us tabulate some of the requirements that we must meet if we are to construct a technique that will represent a model of a system as defined in Def. 3.1.

Postulates for an Ideal Systems Theory Model:

1. Consistency: The model by itself should not introduce illogical factors into the system.

2. Tautology: The model of each element in the system should be identical to, or at least compatible with, the model of the subsystem which is represented by that element. Similarly, a model of a system should also serve as a model of the subsystem for a larger system of which it is a subset.

3. Compatibility: The model should be easily transformed into: (a) a visual or graphical display amenable to human interpretation, and (b) a program easily executable by presently available computers (at least in theory, if not in practice).

4. Exogenous Control: The model should accept exogenous input without having to undergo an extensive modification for each input. In particular, it would be desirable to have a defined model that will respond to continual input for simulation studies.

5. Endogenous Output: The output of the model should be obtainable whenever an input is applied. This includes the particular case in which the output is a null ement (i.e. lack of signal).

6. Flexibility: The model should be applicable to all branches of Systems Theory Studies, especially in Model Synthesis,

Model Analysis, and Decision-Making phases. (More will be discussed on this subject under the Mathematical Considerations.)

Obviously a model should be easy to build and use. It should also be easily adapted to presently available Systems Theory Techniques.

Logical Considerations

"This statement is false." A look at our Def. 3.1 may justifiably provoke criticisms as to its validity as a definition. The definition is tautological since it defines a system in terms of its elements which are also systems. A circular definition is incomplete. However, if we had built a complete definition, we would most likely become inconsistent. The result is the well-known Kurt Gödel's Incompleteness Theorem:

> His theorem states that any adequate consistent arithmetical logic is incomplete, i.e., there exist true statements about the integers that cannot be proved within such a logic (Arbib, 1964).

For our purpose, the above statement about Gödel's theorem should read "logic" rather than "arithmetic logic" and "elements" instead of "integers." The fact that Gödel has used the arithmetic logic for representation is incidental to the essence of his theorem.

Consequently, we may follow a similar line of reasoning for our "system." If we had made our definition about a system complete, then we have to admit that there is a "closed system" that is complete by itself. Then we could build a model that would effectively say: "This System is non-existent," and in so doing we have actually built a model of the system. Assuming that we have not violated a grammatical error, the above statement is a perfectly good English sentence. Since English Language is a system par excellence, the statement is a model of the "system" expressed by itself. If the statement (the system) does exist, then it cannot exist by the power of its own statement; if it does not, then we are denying our own definition. If we argue that English is not a system, then we are denying the definition itself which is in English. Of course, we could also use some other language such as Logic itself: "$A \therefore A = \phi$" or, in English, "If A, then A is a null set," which according to a "complete" system definition should stand by itself.

On the other hand, if we accept the definition we have decided in Chapter II (Def. 3. 1), then we are openly admitting that a system is only a subsystem of another larger system. Now, the statement does not have to stand by itself: "A is an empty set. This system is non-existent." When the previously inconsistent statement is made a part of a larger statement, it can make sense. If not, the process can be repeated until the system will include enough statements to make the sentence a part of a meaningful system.

Of course, we must assume that the structure of a model you are starting with, is not in conflict with the rules you have decided for the consistent modelling. These rules will now have to be discussed. However, before we proceed further, let us comment on Gödel's work.

Gödel. Gödel's first paper appeared in 1931, when he was a 25-year-old German mathematician at the University of Vienna. The article was entitled: "Über formal unentscheibare Sätze der Principia Mathematica und verwandter Systeme, I" (On Formally Undecidable Propositions of Principia Mathematica and Related Systems, I)[8]. The magazine was the German periodical Monthly Mathematical Physics ("Monats, Math. Phys.", vol. 38:173-198) but the text is now available as "On Formally Undecidable Propositions of Principia Mathematica and Related Systems," translated by B. Meltzer, and published from Basic Books, Inc., New York.

The "Principia Mathematica" mentioned by Gödel in his title is an enormous work by Alfred North Whitehead and Bertrand Russell on mathematical logic and the foundations of mathematics (1910, 3 volumes).

The best treatment of Gödel's ideas appears on Nagel and Newman's 118 pages "Gödel's Proof" (N. Y. University Press, 1964) from which the following statements are extracted:

---

[8]Part II was never published.

> In the first quarter of the present century a notable effort was made to put the foundations of mathematics on secure logical foundations, and although few persons were able to follow the recondite processes of reasoning required it was generally accepted in philosophical circles that the theorems of mathematics could all be deduced from a set of axioms with the sole help of principles of logic. In 1931, . . Herr Kurt Gödel, then only twnety-five. . . challenged this belief. Though Gödel's Proof is even more abstruse than the beliefs it calls in question it has convinced those who are able to follow it. . . (Times (London) Literary Supplement).

> . . . Kurt Gödel. . . since 1938 (has been) a permanent member of the Institute for Advanced Study at Princeton. . . When Harvard University awarded Gödel an honorary degree in 1952, the citation described the work as one of the most important advances in logic in modern times (Nagel and Newman, 1964).

After having followed the proof of Gödel's theorem:"If arithmetic is consistent (statement A), there is a true arithmetic statement that is not formally demonstrable in arithmetic (statement G)," or more concisely "A $\rightarrow$ G,"[9] Nagel and Newman say:

> What does this signify? The formula A represents the meta-mathematical statement 'Arithmetic is consistent.' If therefore, this statement could be established by any argument that can be mapped onto a sequence of formulas which constitutes a proof in the arithmetic calculus, the formula A would itself be demonstrable. But this, as we have just seen, is impossible, if arithmetic is consistent. . . We must conclude that if arithmetic is consistent its consistency cannot be established by any meta-mathematical

[9] A implies G.

reasoning that can be represented within the formalism of arithmetic!

Fortunately, the consistency of arithmetic has been proven by Gerhard Gentzen in 1936 using a meta-mathematical principle of "transfinite induction" which is outside the frame of arithmetical reasoning.

Consistency of Tableau. Why are we so concerned about this particular problem? The Tableau method, we are about to develop, is a powerful technique in that it can model systems, ranging from naive models of Chapter IV to more sophisticated models of Chapters VII and VIII and extending to models not contained in this treatise. Its advantage lies mainly in its ability to analyze systems elements into subsets of smaller constituents, and to synthesize (using the terminology of Neumann) supersets using these systems as their elements. As the basic frame work, we have heavily relied on Set and Group Theories, as Arithmetics did, and we will usually be forced to make use of Arithmetics in reaching solutions.

The danger becomes great that we will be seriously tempted to turn a Tableau into a system in which the whole system of deductive reasoning will be incorporated. We start out innocently by using a Tableau to show simple relationships (Chapter IV), but as we progress, we will make use of results obtained from a Tableau analysis to deduce and infer hypotheses of far-reaching consequences. We shall model communication systems and progressively develop a utility hypothesis and so on. We find some interesting relationships within the Tableau that apply correctly when the Tableau is used as a component of a bigger Tableau. We are not restricted in the use of Tableau method to represent any system we can make it fit to model. But we are not guaranteed to yield a consistent result if that system is a meta-mathematical system or contains any reasoning about the systems theory itself. We shall try to keep the Tableau Method to be a general technique, not bound by linearity (unless the system being modelled is considered linear), homogeneity, or other specific requirements not included in the definition of a system. But it is nonetheless a mathematical representation based on the theories of sets and groups, and is bound by their restrictions.

This line of reasoning is directly opposed to Hilbert's proposal of extending his axiomatic approach to a complete formalization of a deductive system.

> The import of Gödel's conclusions is far-reaching, though it has not yet been fully fathomed. These conclusions show that the prospect of finding for every deductive system (and in particular, for a system in which the whole of arithmetic can be expressed) an absolute proof of consistency that satisfies the finistic requirements of Hilbert's proposal, though not logically impossible, is most unlikely (Nagel and Newman, 1964).

Tableau and computers. In formulating our postulates, we have also included a requirement for the model to be executable by a presently available computing machine. In this chapter, we shall show how a Tableau model could be executed by a digital computer. In Chapter VI, we shall show the close relationship between a Tableau and a Turing Machine. Here again, we shall be aware of the implication of Gödel's theorem.

> Gödel's conclusions bear on the question whether a calculating machine can be constructed that would match the human brain in mathematical intelligence. Today's calculating machines have a fixed set of directives correspond to the fixed rules of inference of formalized axiomatic procedure. The machines thus supply answers to problems by operating in a step-by-step manner, each step being controlled by the built-in directives. But as Gödel showed in his incompleteness theorem, there are innumerable problems in elementary number theory that fall outside the scope of a fixed axiomatic method, and that such engines are incapable of answering, however, intricate and ingenious their built-in mechanisms may be and however rapid their operation (Nagel and Newman, 1964, p. 100).

If we omit this "Computerizability" requirement, however, Tableau Method does present some possibility of solving problems that have not been accessible to Arithmetic. A <u>basic</u> Tableau is based on the concepts of elements, sets, mapping and groups, but is not bound by other restrictions imposed upon Arithmetics such as decomposition and factorization rules specified in the "Fundamental Theorem of Arithmetic" (i. e. any integer n greater than 1 can be expressed as a product $n = p_1 p_2 . . . . p_r$ of positive prime numbers, and this expression is unique apart from the order of the factors

(Mostow, Sampson, and Meyer, 1963, p. 59).

The Tableau's counterpart of these rules are the very lose concepts of "fan-in" and "fan-out," terms adopted to avoid any connotation of rules regulating the processes.

Details of this discussion as well as that of how we are planning to avoid the Inconsistency problem will be discussed later in this chapter.

Heisenberg's Uncertainty Principle. The importance of Gödel's Incompleteness Theorem is probably comparable to what Heisenberg has done to Quantum Mechanics with his Principle of Uncertainty:

> It is impossible to devise an experimental procedure for the measurement of x (coordinate of the physical position of a particle) and p (its canonically conjugate momentum) that would yield simultaneously absolutely precise values of x and p: the information provided by any experiment regarding the simultaneous values of x and p is always inaccurate to such an extent that $\Delta x \Delta p \geq h/4\pi$ ($\doteq h/2$). (h = Planck's constant = 1.05×10^{-34} joule-sec.) (Rojansky, 1938, p. 122).

We have already discussed the example of "Maxwell's Demon" by Wiener based on this principle. The important thing to notice in both cases is that our presently available techniques do not allow us to make any statement about these facts. They do not make any assertion about the actual outcome.

Heiseberg's principle does not prevent a particle that will pass by a point at x to have a momentum p. Gödel does not say that the truth of a system and its consistence are wrong or cannot be proven. He is merely stating that the present system, by itself cannot make a statement on itself and be justified by its own system of logic. This does not necessarily make the system wrong. The fact that the probability of any particular event occurring at a certain time is zero does not prevent it from actually happening then.

Let us next examine postulate 3 before returning to other mathematical considerations.

Systems Representation

Graphic Considerations

Black box. A black box representation is probably the most popular graphic representation used in systems theory today, and has the obvious advantage of being suitable for representing a tautologous system as defined in Def. 3. 1. It allows a black box to become an element of a larger black box, and so on.

Let us assume that we have identified n elements of a system S, and we have recognized the particular mapping operations that exist variously among each group of those elements, between the groups taken as subsets, and in relation to the external environment, i. e. exogenous inputs and endogeneous outputs. The observation that

x_2, one of the n elements, is related to another element x_1, by an operation (x_2, x_1) mapping x_1 into x_2, can be expressed in a functional form as: $f(x_2, x_1) = f(x_2/x_1) = f_{21}(/x_1)$ or simply, $= f_{21}(/)$. Not knowing anything about the true nature of the dependency, we are only allowed to state that the operation of x_2 depends on the condition that x_1 is present.

On the other hand, if the relation did become more precisely known later, we could either substitute the true relationship for the /, or state the function near the black box.

When left undefined, $f_{21}(/)$ has no more meaning than that afforded by the theory of mapping (a topic we shall discuss more in detail later in this chapter) and may be interpreted as an expression of the relationship of a physical, chemical, sociological, economic, mathematical, or other nature. For example, it may mean that x_2 is socially dominated by x_1; x_2 is a temperature 10° C below that of x_1; or x_2 is the risk involved in a plane trip of x_1 miles. To keep our example simple, let us assume that f_{21} is a single-valued function that, say, amplifies every input x_1 by a factor of two.

$$x_2 = f_{21}(/x_1) = f_{21}(2^*x_1) = 2^*(x_1)$$

Obviously, an alternate method for describing $f_{21}(/)$ may be a complete listing of mapping between x_2 and x_1, not unlike the tables provided for such irregular (if not illogical) mapping functions as taxes.

x_1	0	1	2	3	4	5	6	...
x_2	0	2	4	6	8	10	12	...

The black box notation will be used directly, except that we shall make it contain a / in the box (or the function if it be known exactly), and include with it or replace it by a parameter of transformation if there is no ambiguity (e.g. simple addition or multiplication by a scalar where the operations are obvious). Also, we require the input flow to be perpendicular to the output flow with a usual (but not absolutely adherred) practice of arrows pointing in counter-clockwise direction. Various possible presentations are shown for our simple amplifier in Figure 3.1.

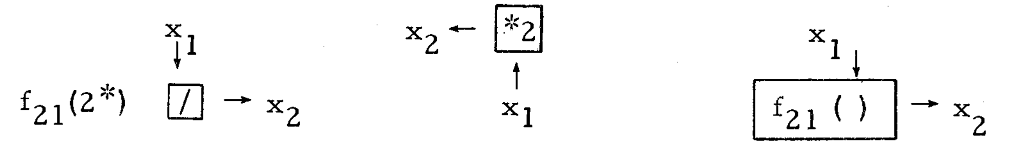


Figure 3.1. $x_2 = f_{21}(x_1) = 2^*x_1$.

<u>Transducers.</u> A transformation is therefore indicated by a perpendicular shift in direction from the primary input flow to the transverse (transposed) output flow. An input flow through a black box without change in direction does not alter the nature of the input. In other words, we are assuming that the information flow can have as many "fan-outs" as one may wish and that the information input received by each black box is identical to that received by any other

on the same line. Of course, part of the message may be lost by the transformation that is indicated by the particular box. One such element of a particular importance is a "Threshold Element" that will yield either a binary "1" when the input signal is larger and a binary "0" when the input is equal to or less than the threshold value of θ. This binary detector is the same as the Threshold Logic we have described in Chapter I (page 22).

Several output flows may be joined together to affect the dependent variable. $x_3 = f_3[f_{31}(x_1), f_{32}(x_2)]$ shown below is such an example. The nature of the "fan-in" function f_3 must be known to compute x_3.

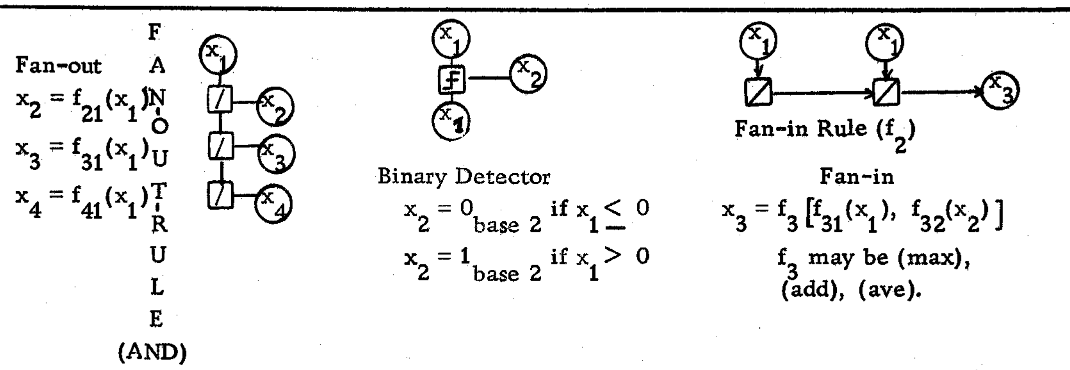


Figure 3.2. Fan-in, fan-out, and binary detector.

State variables. Since we are defining the system as a set of elements, we should attribute to these elements semi-permanent characteristics. We shall call these elements the "states" of the particular system. Therefore, there will be as many states as there are elements in the set or subset being considered, and the complete knowledge of all these states (a state vector) at any instant will provide a complete snap-shot of the system at that instant. Each state at a time t is a function of its former state and the input received since then. Unless otherwise stated (this is a very unlikely occurrence) the state will, therefore, act as a temporary memory, or a register, retaining its state condition until an input has changed it to a new state condition. In order to avoid the possible misinterpretation, whenever there is a possibility for ambiguity, we shall speak of "state variables" as the elements, and "state value" (or eigen-cell value) as the numerical expression of the state condition.

State variables will be shown either as a box with a main diagonal (opposite of the /), or as a circle. We shall also try to have them arranged such that their main diagonals will form a single line (Figure 3.3). Such a line is called an eigen-line, while the cell or the circle representing the state is called an eigen-cell.

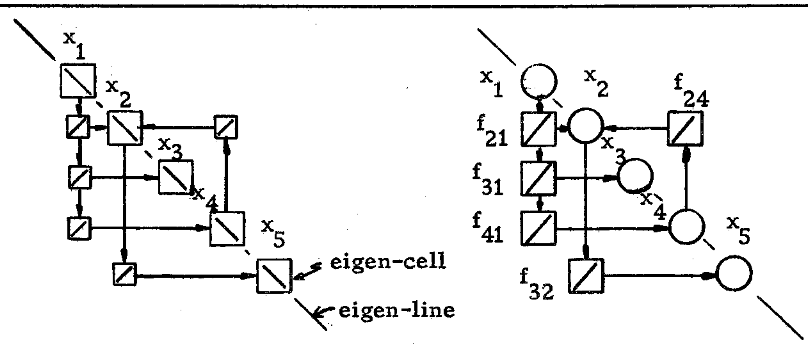


Figure 3.3. States representations.

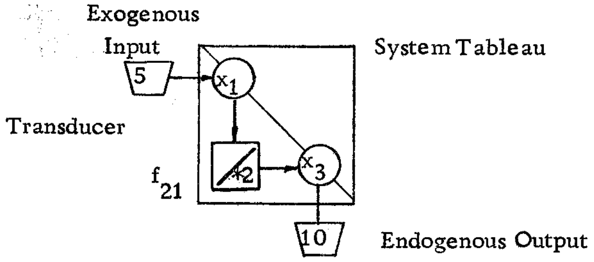


Figure 3.4. Endogenous output, exogenous input.

Chained transducers. Several operations (mapping, transducer, function, operator, ... are all synonyms) may be chained, or cascaded to form a single path. An example of a chain formed by four of our simple "doublers" is illustrated below in Figure 3.5.

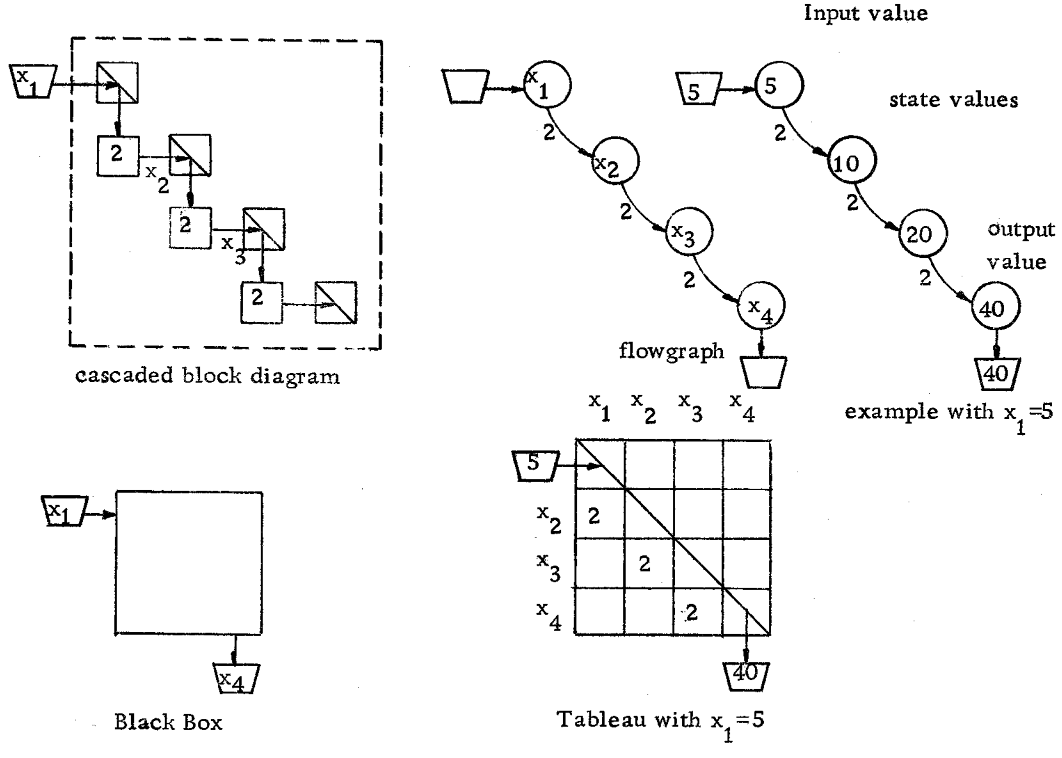


Figure 3.5. Various representations with a cascaded transducer.

Flowgraph and Tableau. From this point, it is easy to see how a network approach (a flowgraph) or a matrix approach (a tableau) may be produced for further analysis. Figure 3.5 showed one such example, and Figure 3.6 illustrates many varieties of configurations that commonly occur. Clearly, a Tableau is the easiest to construct when the relations are most complex, and a flowgraph can be generated directly from it later.

Programming Considerations

Network structure. Our Tableau method is slowly beginning to take shape as the postulates are becoming satisfied one by one. Perhaps, this is the time to examine its structure to see whether our method is basically compatible with the fundamental structure of a modern digital computer (3b, page 72).

We have not yet decided on the nature of mapping functions themselves and we shall try to preserve this generality as far as possible.

Let us assume that we are given a system composed of five elements: a, b, c, d, and e. "e," the endogenous output of the syste, is a function of the four other variables. The function is then decomposed into binary operations using the associative axiom by assuming that five elements compose a group. (If such were not the case, a smaller group will have to be found first.) Let's formulate

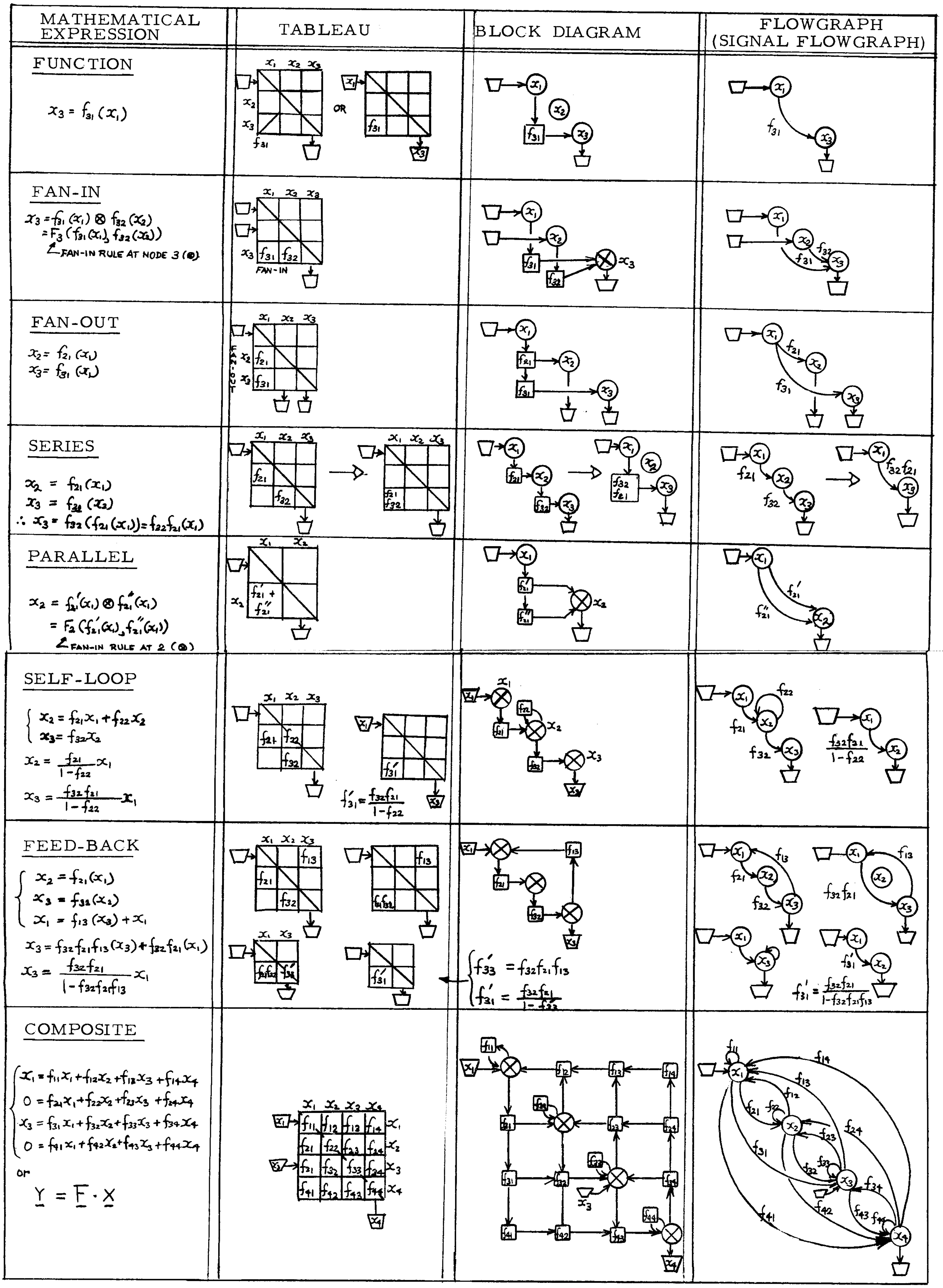


Figure 3. 6. Basic configurations.

this as an example.

Example 3.1. a and b are "mixed" into a product (say x_1), c is mixed in with x_1 to produce x_2, then d and x_2 yields x_3 which is again mixed in with a as x_4, and finally the result x_4 is taken out as an endogenous value e.

Polish notation. The so-called Lukasiewicz, Polish, or parenthesis-free form is a "right" or "left" list of an arithmetic or logical statement. By this, it is meant that all dependent variables are listed to the left (or right if left-listing is used) of the operation to which it pertains. For example, a+b will be written ab+, a union with b may be written as abU, etc. The independent variables are usually grouped by two (or less), followed directly by the operation relating the two to produce the dependent variable.

For example the statement (a system) of Example 3.1 can be written as:

$$\begin{array}{ccccccccccc} a & b & \boxtimes & c & \boxtimes & d & \boxtimes & a & \boxtimes & \rightarrow & e \\ & & x_1 & & x_2 & & x_3 & & x_4 & & \end{array}$$

The symbol $\boxtimes$ is the same "mixer" symbol used since Chapter I and denotes an unknown mapping corresponding to the / of our graphical notation. We have named them for our convenience to be: x_1, x_2, x_3, and x_4.

Another example of this notation may be:

ab + [(d+e)a + (d+c)b)e + gf] c which will be written as:

$$ab^{*}cde + a^{*}bcd + * + e^{*}fg^{*} + * +$$

where * indicates a multiplication (a notion that we shall continue using in preference to x) of the preceding two terms, and + indicates the addition of the preceding two terms.

This notation was first proposed by Jan Lukasiewicz in Polish and was subsequently translated into English in 1951 as "Aristotle's Syllogistic from the Standpoint of Modern Formal Logic," and finally into American as "Elements of Mathematical Logic" in 1963 (McMillian).

Tableau in Polish notation. Figure 3.7 shows a Tableau constructed for ab⊠ and Example 3.1. It may be noted that independent

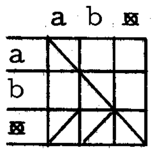


Figure 3.7.a. Polish Tableau

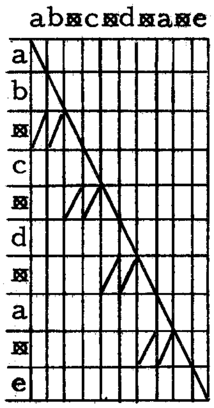


Figure 3.7.b. Polish Tableau

variables have no / in their respective columns (a, b, c, d, and a), while the dependent variable (end product "e") has no / in its row. Also the familiar staircase pattern can be recognized relating all operations into a single path. The details of these mechanisms are similar to those described in prior works written for CPS Tableau (Riggs and Inoue, 1965; Riggs, 1966; Inoue, 1964).

Example 3.3.

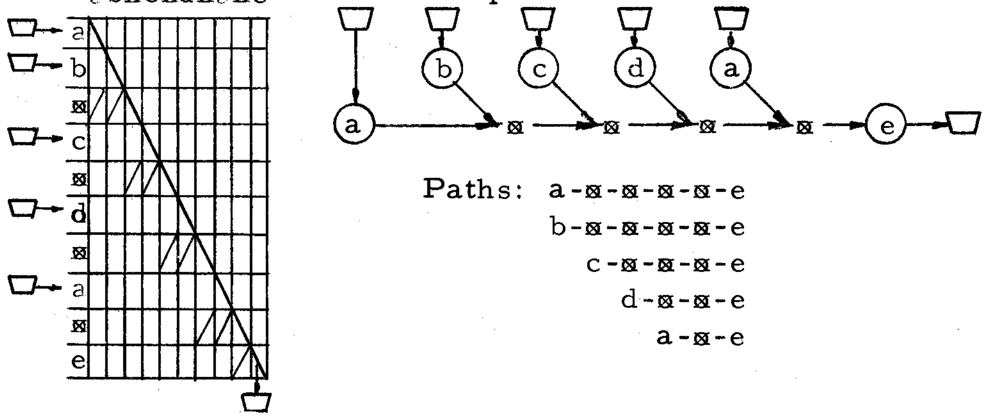


Figure 3.8. A single chain of operations.

Macro-programming

An immediate application of this Polish notation Tableau is a simple machine language programming corresponding to the single path obtained above in Figure 3.8.

Let us suppose that a computer has a three-address machine language instruction format as shown in Figure 3.9. It would then be necessary for a higher level language such as FORTRAN or Iverson's Programming Language (corresponding to our Polish notation) to be converted into this machine language format before it can be executed by this particular computer. An algorithm that will be used for such a conversion is called a "compiler" algorithm.

We shall have a more detailed discussion on the nature of what should constitute an algorithm later (Chapter VI). For the time being let us accept the definition of algorithm as was provided by Professor Harry E. Goheen (Inoue, 1964): "An algorithm is an automatic scheme for manipulating data, guaranteed to yield solution."

Operation	Address 1	Address 2	Address 3

Figure 3.9. A three-address instruction format.

In the instruction format given in Figure 3.9, a macro-statement such as ab + c - d * a * — e will be executed as a series of machine instructions involving:

Operation: a code specifying the particular operation to be performed.

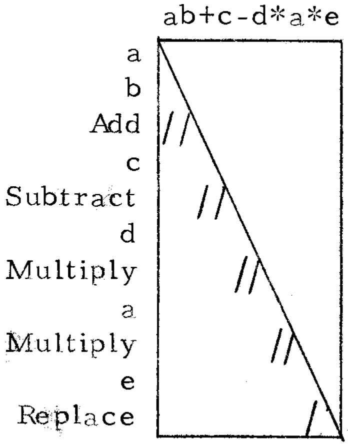
Address 1: the location where the independent variables has been stored.

Address 2: the location of the modifier.

Address 3: the location where the result of the operation should be placed.

In many computers, including the IBM 1620, the third address is omitted. In most operations (at least in additions and subtractions) the results are automatically placed in the same address as the one specified by address 1, replacing the independent variable.

The construction of the Tableau will provide all four pieces of information at once. The operation code is indicated either inside the corresponding cell or on the outside of the grid as we have shown. The two addresses are marked by the /'s and the third address is obtained by following the single-path as we have done in Figure 3.8 (Example 3.1).



Operation	Add. 1	Add. 2	Add. 3
Add	a	b	1
Subtract	1	c	2
Multiply	2	d	3
Multiply	3	a	4
Replace	4	-	e

Figure 3.10. Micro-programming for Example 3.3.

A more sophisticated application to computer programming will be shown later by using the Tableau to replace Hellerman's Tableau. However, it will be more convenient to delay this discussion for the time being and go on to other topics first. What we have done, meanwhile, is to assure ourselves of a seemingly high compatibility between our Tableau method and modern digital computers.

Synthesis and Analysis Considerations

Mathematical Considerations

Group theory. It is rather difficult to discuss mathematical relations without bringing in linearity as we have discussed in Chapter I. On the other hand, it is very important to build the Tableau method so that its use will not be exclusively limited to linear models. Rather, we would like to have a method which will help us start modelling a system even while we are waiting for concrete data to develop so that more exact relationships between variables can become known.

Of course, we are also hoping that the same model will lead us, through gradual improvements and refinements, to a highly sophisticated model in which the effects of noise, distortions, and competitions can all be included stochastically to permit continuous updating of our policy and decisions in a practical manner (see Chapters VI for examples). Then we hope that an automatic learning or adaptive process can be incorporated, so that the model will only ask management's help when exceptional decisions (i. e. non-routine environment or endogenous changes) occur.

Meanwhile, we are still at the very beginning, trying to build a structure for a model of rather nebulous character. Let us turn to Group Theory to see what is available in line of hardware!

Theorem 3.1. Identity element and inverses. Let G be a group (Def. 1.13, p. 29) with binary operation $\boxtimes$ (Def. 1.7, p. 12). Now, (1) if $a \boxtimes c = a$ or $c \boxtimes a = a$ for some $a \in G$, then c must be the identity element e of G; (2) if $a \boxtimes c = b \boxtimes c$ or $c \boxtimes a = c \boxtimes b$ in G, than $a = b$; (3) if $a \boxtimes b = e$ or $b \boxtimes a = e$, then $a = b^{-1}$ and $b = a^{-1}$.

Proof of Theorem 3.1. (1) Let a^{-1}be the inverse of a (from Def. 1.13 (2), page 29, $a \in G$ must have an inverse). Then, $a \boxtimes c = a$ implies: $a^{-1} \boxtimes (a \boxtimes c) = a^{-1} \boxtimes a$. Since $\boxtimes$ must be associative (page29), $(a^{-1} \boxtimes a) \boxtimes c = (a^{-1} \boxtimes a) = e$, or $\underline{c = e}$. Similarly,

$$c \boxtimes a = a \Rightarrow (c \boxtimes a) \boxtimes a^{-1} = a \boxtimes a^{-1} \Rightarrow \underline{c = e}. \quad \underline{\text{Q.E.D.}}$$

Note:
- $\in$ stands for "an element of" or "contained in."
- $\Rightarrow$ means "if, then..."
- $\therefore$ means "therefore"
- $::$ means "identically equal to"
- $\Leftrightarrow$ or "$\rightleftarrows$" means "if and only if," or iff.

(2) By the same token: $a \boxtimes c = b \boxtimes c \Rightarrow a \boxtimes c \boxtimes c^{-1} = b \boxtimes c \boxtimes c^{-1} \Rightarrow a = b$

$$c \boxtimes a = c \boxtimes b \Rightarrow c^{-1} \boxtimes c \boxtimes a = c^{-1} \boxtimes c \boxtimes b \Rightarrow a = b$$

(3) Also, $a \boxtimes b = e \Rightarrow a \boxtimes b \boxtimes b^{-1} = e \boxtimes b^{-1} \Rightarrow a = b^{-1}$: others follow.

Theorem 3.2. Uniqueness of solutions. Let G be a group with $\boxtimes$. The equations: $a \boxtimes x = b$ and $y \boxtimes a = b$ have unique solutions for x and y, $x = a^{-1} \boxtimes b$ and $y = b \boxtimes a^{-1}$ for any two $a \in G$ and $b \in G$.

Proof of Theorem 3.2. $a \boxtimes x = b \Rightarrow a^{-1} \boxtimes a \boxtimes x = a^{-1} \boxtimes b \Rightarrow x = a^{-1} \boxtimes b$. Now, if x and x' were two distinct solutions, then: $a \boxtimes x = b$ and $a \boxtimes x' = b \Rightarrow a \boxtimes x = a \boxtimes x' \Rightarrow$ [from Th. 3.1 (2)] $\Rightarrow x = x'$

A similar proof will show the y to have a unique solution. If G were abelian (commutative), then $x = y$.

<u>Theorem 3.3. Source of groups</u>. Let S be an arbitrary set, and let M denote the set of all one-to-one mapping of S to itself. Then M, with composition of mappings as a binary operation is a group. Its identity element e is the identity mapping of S, the inverse of an element f of the group M is the inverse mapping f^{-1}.

<u>Proof of Theorem 3.3</u>. (This is one of the theorems that made Tableau method possible.) Let f, g be two elements of M, that is, two one-to-one mapping $S \rightarrow S$. Then the operator (fg) will be the mapping that will send an arbitrary element $x \in S$ into $f[g(x)]$. Since $f[g(x)] = (fg)x$, (fg) may be considered as a one-to-one mapping and must therefore also be an element of M. Thus, the composition of mapping such as the one producing (fg) from f and g, is a binary operation in M. Now let us see whether M qualifies as a group (Def. 1.13, page 29).

First, as to the associativeness: let f, g, $h \in M$ and $x \in S$, then f(gh) sends x into $f[gh(x)] = f[g(h(x))]$. Similarly, we have (fg)h sending x into $(fg)h(x) = fg[h(x)] = f[g(h(x))]$. Thus, $f(gh) = (fg)h$ and this satisfies the associative axiom (Def. 1.11, page 28).

Second, as to the identity element: let e be the identity mapping of S, or, $e(x) = ex = x$ for all $x \in S$. Then, $fe(x) = f[e(x)] = f(x)$ and $ef(x) = e[f(x)] = f(x)$ imply that $ef = fe = f$ for any f in M makes e an identity element of M (Def. 1.10, page27).

Third, as to the inverse: let f^{-1} be the inverse mapping of f. Then, $f[f^{-1}(x)] = x = f^{-1}[f(x)]$ for any $x \in S$. Therefore, $ff^{-1} = f^{-1}f = e \equiv f^{-1}$ is the inverse for binary operation as well.

All conditions are satisfied and M is a group with identity element e and the same inverses as S. Q.E.D.

Definition 3.2. A subgroup. Let G be a group with operation $\boxtimes$, and let H be a non-empty subset of G. Then H is called a subset of G if for any elements a, b in H the elements a^{-1}, b^{-1} and $a \boxtimes b$ are also in H.

Theorem 3.4. Identity element in a subgroup. Let H be a subgroup of a group G. Then:

(1) G and H have a same identity element.

(2) The inverse of an element in H is the same as the inverse of the same element in G.

(3) The identity element of any group forms a subgroup by itself.

Proofs of Theorem 3.4. (1) By Def. 3.2 H is not an empty group and therefore must contain at least one element "a." Moreover, Def. 3.2 requires that its inverse a^{-1} is also an element of H, and the binary operation $\boxtimes$ also belongs to H as well as to G. Thus, $a \boxtimes a^{-1} = e$ must be an element of both H and G, and must be unique according to Theorem 1.3 (page29).

(2) This follows directly from the proof (1) above. The element e is unique in both G and H, and from Theorem 3.2 (page 96) the solution of the equation: $a \circledast x = e$ is unique and is equal to $x = a^{-1} \circledast e = a^{-1}$ in both G and H.

(3) The identity element e by itself meets all the requirements for being a subgroup as specified by Def. 3.2. e, an element of H, its inverse $e^{-1} = e$, and $e \circledast e = e$ are all in H.

Model Synthesis

Group Transformation Table. We have demonstrated that the mapping of a group may itself become a group (Theorem 3.3) and that elements of a group may be a subgroup with all the characteristics of a group. Let us now examine the relationship that exists between a Group representation and a Tableau representation of the previous section.

The matrix of binary operation may now justifiably be called Group Transformation Table. Other popular names are Permutation Table, Group Multiplication Table (Stephenson, 1965), etc. but they all seem to have narrower connotation (for example, one hesitates to say "a multiplication table for a binary operation of addition").

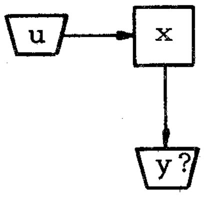
Let us look at an example of a reduced Flintstone example with four fingers e, a, b, and c. The transformation table is shown in Figure 3.11. Note that we are essentially looking at a black box with input u and state x and trying to examine the output y. The transformation table tells us the results of our "model analysis."

State (x)

Input (u)	e	a	b	c
e	e	a	b	c
a	a	b	c	e
b	b	c	e	a
c	c	e	a	b

Output (y)

Transformation Table



Model Analysis

Figure 3.11. Transformation table for model analysis.

<u>Isomorphism</u>. Let us illustrate some examples of isomorphism. Aside from the usual 0, 1, 2, 3 interpretation of e, a, b, c, with the binary operation of addition, we can show two more examples that are also isomorphic with respect to Flintstone's four fingers. The first one is the complex number set 1, i, -1, -i with its binary operation of multiplication. This may also be interpreted in the complex plane as a set $\{e^{o}, e^{i\pi/2}, e^{i\pi}, e^{i3\pi/2}\}$ of vectors cycling around the origin. The second example is a set of four matrices: $\left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \right\}$ with matrix multiplication as its binary operation. These two sets satisfy the requirements for being groups under Def. 1.13 (page29) and also for being isomorphic under Def. 1.15 (page32).

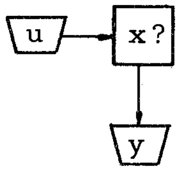
<u>Tableau</u>. Before a model can be used for analysis, it must first be built from our observation of the actual system (page43).

What we need therefore is a table that will give us the state x (the internal condition of the system) for each combination of an input (u) and an output (y). For our example, such a table is shown in Figure 3.12.

Output (y)

Input (u)	e	a	b	c
e	e	c	b	a
a	a	e	c	b
b	b	a	e	c
c	c	b	a	e

State (x)



Model Synthesis

	e	a	b	c
e		c	b	a
a	a		c	b
b	b	a		c
c	c	b	a	

Figure 3.12. Tableau and exchange table for model synthesis.

When we connect the identity elements by a line, we suddenly realize that the table we have constructed is in fact what we have been calling a Tableau. The internal state that gives a particular output for a given input is in fact a transducer. What we have been calling an eigen cell is the identity element, and the eigen line is the line we used in connecting all our identity elements.

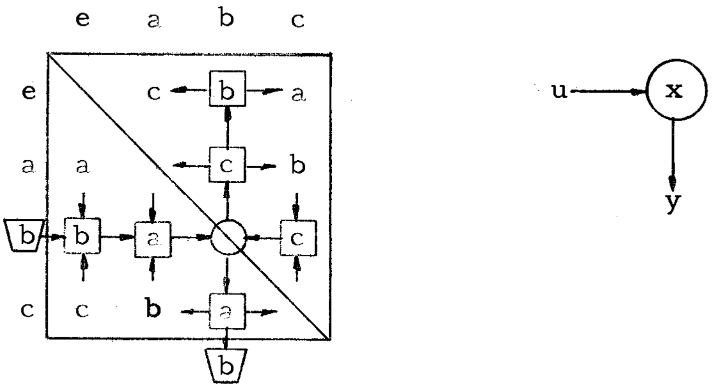


Figure 3.13. Interpretation of an eigen cell.

We should also test our newly found Tableau to see whether the exogenous input and endogenous output relations do hold for each eigen-cell as well as for the entire tableau. In Figure 3.13 we examine the cell b and note the fan-in from three transducers b, a, and c on the same row, meaning "b state may be reached either by an exogenous input b outside the tableau, the output of state e transformed by b, the output of state a transformed by a, or the output from state c transformed by c." Similar interpretation can be given to all the transducers and endogenous output b, on column b as the state b may be fanned-out to.

Definition 3.3. A Tableau. A Tableau of dimension n is an nxn matrix representation of a set of n elements such that an entry in a cell a_{ij} or row i and column j expresses the mapping of the ith element into the jth element. In particular, the cells corresponding to elements a_{ii}, i = 1, ... n are called eigen-cells and are transversed by a main diagonal called eigen-line.

In this chapter, we have tried to establish Tableau Method as a generalized Systems Tool with as little constraints as possible that may hinder its application, present or future, in the study of any system meeting our Definition 3.1. To escape from the usual assumption of linearity (linear matrix models), specialized fan-in and fan-out conventions (AND-AND in CPS, OR-OR in signal flowgraphs, etc.), homogeneity (Kirchoff's Laws, Lagrange's conservation of energy, etc.), commutativity [the so-called Abelian groups, after the Norwegian mathematician N. H. Abel (1802-1829)], and many others, we have chosen the most basic mathematical language (theories of sets and groups) and the most basic graphical representation (block diagram) as the two bases for the Tableau Method.

As far as Gödel's Incompleteness dilemma is concerned, we shall try to build our method with three safeguards: (1) avoidance of any meta-mathematical, or meta-systematic statement:

> The rule that is most often adopted by mathematicians is to declare that the totality of sets is not a set. Even more, one never uses 'all the sets such that...,' but rather 'all the subsets of the set A such that...' With such conventions, the Russell paradox ('y is not an element of y' by Bertrand Russell in 'Principia Matematica') cannot be formulated, and we escape, as far as we know at present, the spectre of contradiction in logic (Mostow, Sampson and Meyer, 1963, p. 510).

(2) Adoption of <u>Logical</u> Tautology (the truth of the output depending entirely on the truth of the input) which has been proven successfully for its consistency (and its incompleteness) by Russell, whenever it is consciencioulsy possible to do so, and (3) denouncement of any applicability of Tableau method to problems dealing with reverse flow of time (including Minkowski space problems).

IV. STRUCTURE TABLEAU

> ... we trained hard -- but it seemed that everytime we were beginning to form up into teams we would be reorganized -- I was to learn later in life that we tend to meet any new situation by reorganizing, and what a useful method it can be for creating the illustion of progress while producing confusion, inefficiency, and demobilization -- but what fun [Petronius Arbiter and Schwartz, written on a clay tablet, 210 B.C. (Schwartz, 1966)].

Studies of Structures, Relationships, and Organizations

Sociograms

The counterparts of flowgraphs in engineering, are called graphs of social relations, or sociograms. In the order of strength, we may roughly classify them as expressing three types of mapping:

1. Sociometric relations, where preferred choices of individuals are expressed. For example, John wants to sit next to Mary, but Mary would rather sit with Mike, etc. A simple example is shown in Figure 4. 1 with an equivalent Tableau example.

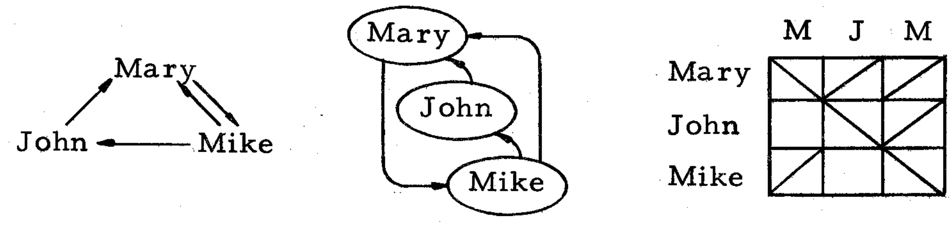


Figure 4. 1. Sociometric relations.

2. Communication relations, where communication links between individuals or groups of individuals are shown. For example, Smiths communicate with Jones and Millers, but Millers and Jones are not speaking to each other. Notice that a communication relation requires a two-way link rather than a one-way or two-way relation permitted in sociometric relation Figure 4.2 illustrates our example.

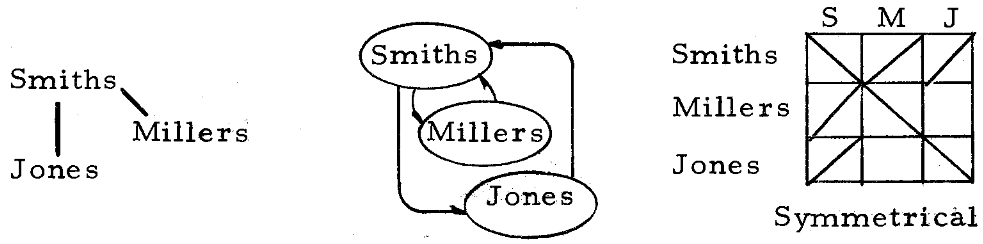


Figure 4.2. Communication relations.

3. Dominance relations. This is where a social hierarchy is established. For example, Mr. Smith is Mr. Miller's boss, but Mrs. Miller dominates Mrs. Smith, etc. In contrast to the communication model, a dominance model does not allow a two-way link between any individual or organization and another. Figure 4.3 illustrates such an example.

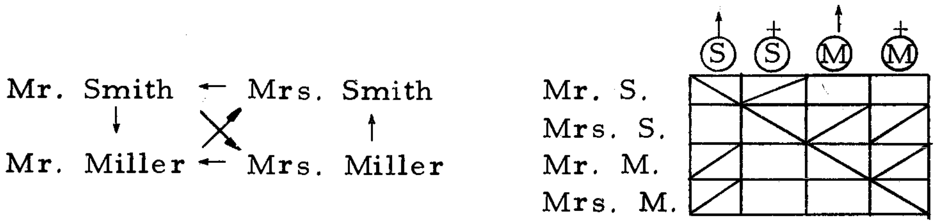


Figure 4.3. Dominance relations. Note: Either a_{ij} or a_{ji} must be entered for each pair (i, j).

Exchange Models

A similar model in Economics is called an exchange model. The model may be used to show the flow of goods between various departments of an organization, between various organizations in an industry, between industries in a country, or imports and exports between countries. Figure 4.4 shows an example of trades between several countries (fictitious data).

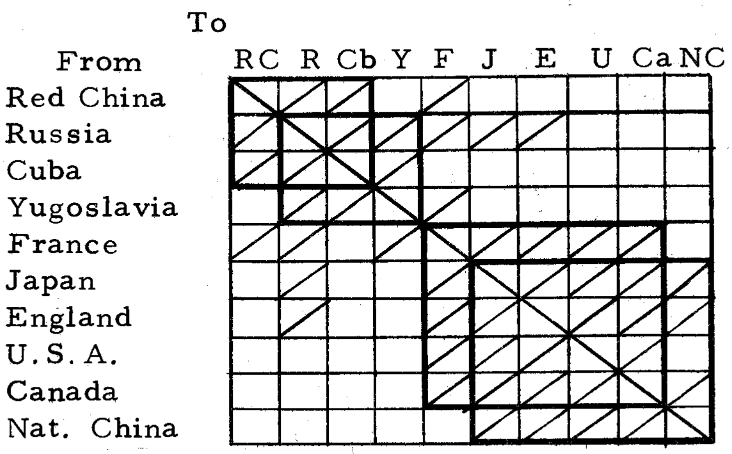


Figure 4.4. Exchange model.

Business Application

A similar matrix is used in accounting to show the services rendered by a department to other departments so that each department may be charged for the actual amount of benefits received from other departments. However, this matrix will be of no actual value until numerical figures are inserted instead of mere recognition of relationships (/). Therefore, we shall not include it under the structure tableau category.

Engineering Applications

Engineering applications are very numerous and varied in nature. We shall only illustrate few examples.

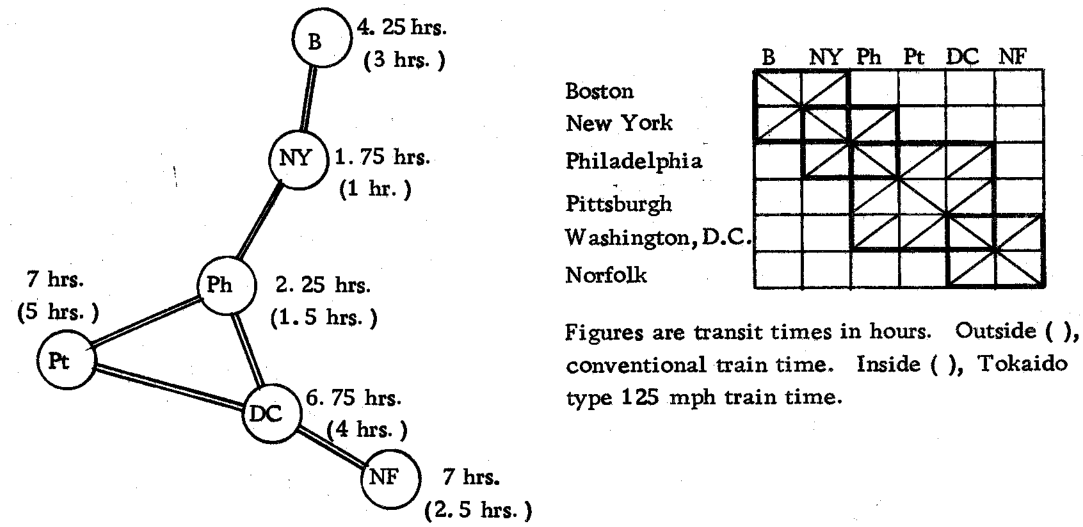
Parts explosion model. When a product is assembled, it is likely that several components are used at many stages of its assembly. Screws, paint, electronic subassemblies (micro-module circuits), rubber gaskets, etc., are some of the examples. A relation matrix is usually employed to show the relationships between those components and subassemblies and the final assembly.

Output Parts and assemblies.

Input Parts and Assemblies	A	B	C	D	E	F
A	\			/		
B	/	\	/			
C	/		\	/		
D				\		
E			/	/	\	
F					/	\

Figure 4. 5. Product-explosion tableau.

Transportation model. TRW Systems (Company address: 1 Space Park, Redondo Beach, California) have used a matrix approach in finding the best paths of transportation between northeastern cities. This is essentially the same as the Communication models in sociology, and results in a Tableau that is symmetrical with respect to its eigen line. Figure 4. 6 shows an illustration.



Figures are transit times in hours. Outside ( ), conventional train time. Inside ( ), Tokaido type 125 mph train time.

Figure 4. 6. Northeast corridor transportation.

The use of this type of Tableau will be very helpful in formulating the problem before reliable quantitative data become available. Often a model is needed precisely for the sake of gathering data (chicken and egg, modern version) and obtaining public support for further study. Let us quote from Frankel and Gilon (1966) who worked on this particular project.

> Without a clear understanding of the issues involved in the transportation problem, it is futile to propose solutions. For example, considerable attention is being paid to 'high-speed trains' as a result of the the initial success of the Tokaido Line in Japan. This success has spurred U. S. attention towards a 125 mph train for the Northeast Corridor. The introduction of such a train will undoubtedly decrease travel time . . . New York to Washington, 2. 5 hours vs. current 4 hours . . . Estimates indicate that the decrease travel time between New York and Washington

> will cost approximately 2.5 million dollars per minute. ... The real issue however is not in the price per minute saved but in the public acceptance of such a solution. Without public support the solution is an academic one (Frankel and Gilon, 1966).

Data Bank. When Lockheed Missiles and Space Company (a group division of Lockheed Aircraft Corporation, Sunnyvale, California) was requested to conduct "California Statewide Information System Study" (for $100, 000) in 1965, one of the first things the company did was to construct a data interchange matrix, part of which is shown on Figure 4.7.

Based on this, the company has finally recommended an Information Center (Data Bank) for the sources of information available in the State of California. This center, unlike those usually proposed by other projects, acts only as a clearing house and also as a library card catalog that will serve the users to look up where the information is stored and connect them to these sources, much like a telephone exchange system. Figure 4.8 illustrates this concept.

Critical Path Scheduling. In order to render our presentation complete, we must include few words on the use of Tableau as a matrix representation of arrow network. Unfortunately, when the Tableau method was developed for Critical Path Scheduling, we had no idea of the potentiality of Tableau method for other uses but C.P.S. Because of this particular application, we had, at that time,

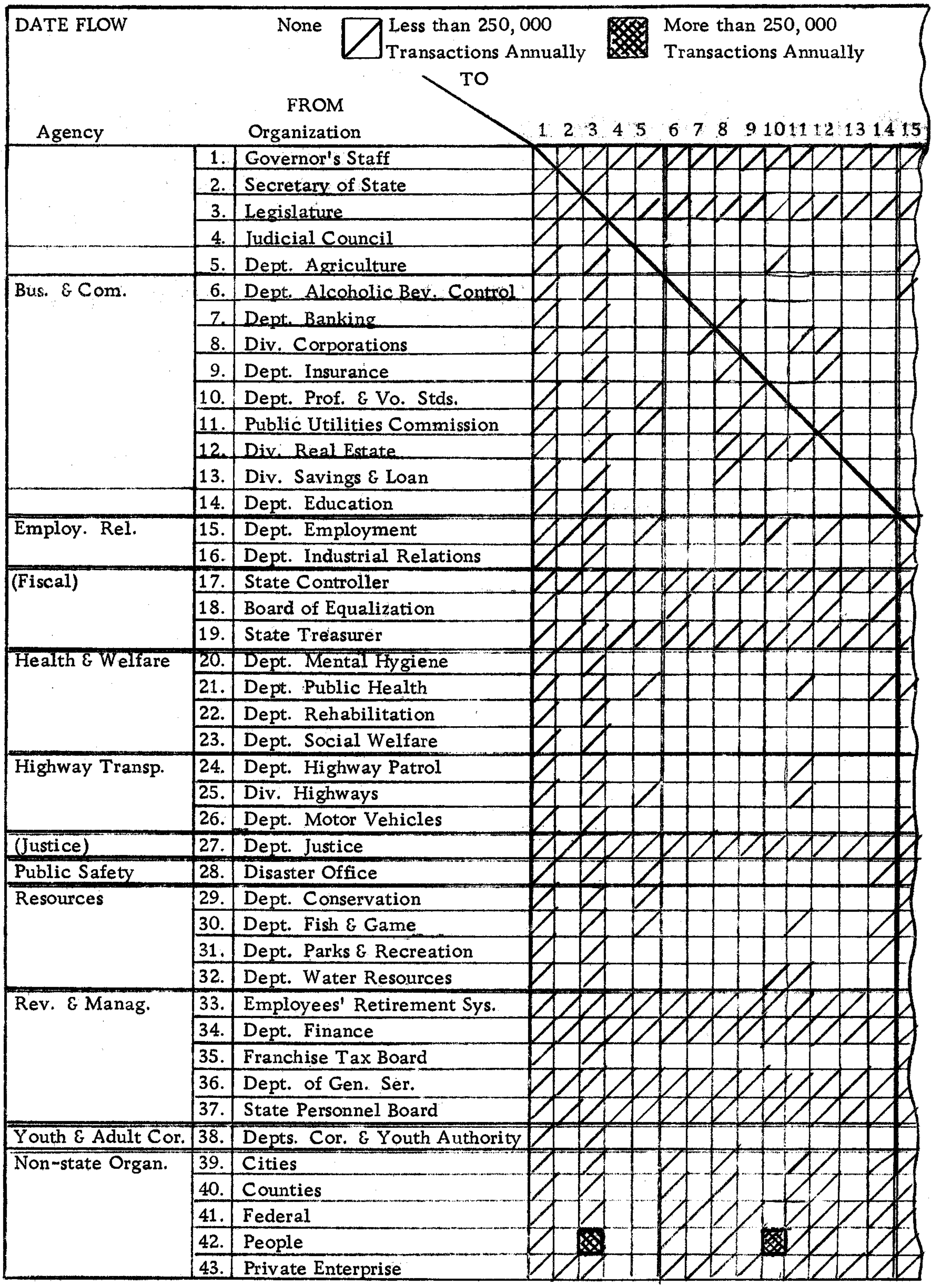


Figure 4.7. Lockheed data interchange tableau.

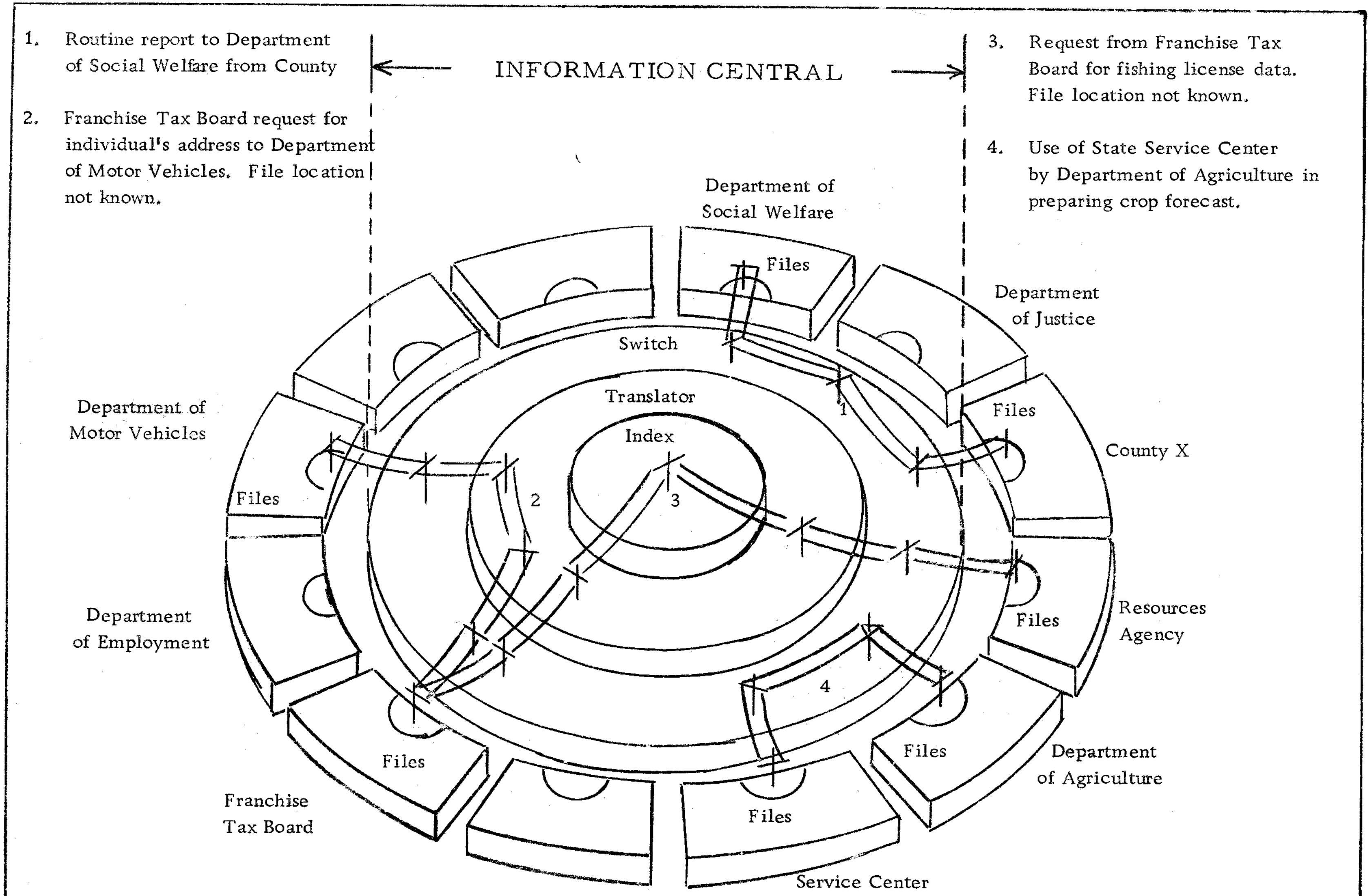


Figure 4. 8. Coordinating role of information central.

adopted the transpose of what we now consider the proper representation of a Systems Tableau.

The decision to adopt the present representation as the proper one, rather than the one we had advocated for Critical Path Scheduling was a difficult decision to make. However, the readers who have already examined several models used by various people in various professions, such as those illustrated in this chapter, will probably agree that our decision is probably the most plausible one. The orthodoxity of this representation will become even more clear when we examine Tableau Économique and Simplex Tableau. In this era of Western culture, we have adopted the habit of writing equations horizontally. This fact which accounts for this particular representation, would have been completely different if modern technology were developed in China or Japan, where conceivably equations would have been written vertically from top-to-bottom. We may even amuse ourselves by considering the Simplex Dual in the light of Kipling's "Oh, East is East, and West is West, and never the twain shall meet" (except upon a Tableau!).

This paragraph is also presented to our readers as somewhat of an apology and justification for our dwelling on applications and theories that seem at times tedious, trite, and trivial. This may be felt even more seriously later, in view of the fact that the latter part of this treatise will have to be greatly condensed and many ideas will have to be presented without proof. However, we feel that being careful in the foundation stage is the only possible chance to avoid committing sad discrepencies of fundamental nature. This is also the reason why we are relying very heavily on the Set Theory and Group Theory, and even in these, are trying to follow closely such reputable authors as Mostow, Sampson (whom I had the privilege of having as an instructor), and Meyer.

Fortunately, our choice of primary matrix is not detrimental to any of our analysis. To show how the critical path scheduling problem looks under the new orientation of the Tableau, we present the (perhaps overly quoted) example of Patio Construction by Riggs and Heath (1963).

> Patio Example. The vacation cabin is situated 100 miles from the owner's home. He plans to build a protected, covered patio at the cabin. Supplies and handyman labor are available not far from the cabin (Riggs and Heath, 1963, p. 7).

Figure 4. 9 is the Tableau corresponding to this project. The symbol D placed over the / indicates a dummy connection the implication of which will be discussed presently. From the network in the right top of Figure 4. 9, we notice that an arrow network is composed of arrows depicting the states but transducers are completely ignored and are understood at the junctions of the arrows. This convention works fine if all the fan-ins are identical for all the fan-outs, and all the fan-outs are identical for all the fan-ins. In other words, the

		A	B	C	D	E	F	G	H	I
Lumber	A									
Blocks	B									
Excavate	C									
Foundation	D		/	/						
Frame	E	/			/					
Decking	F					/				
Roofing	G					/				
Clean up	H						/	/		/
Windbreak	I		/							

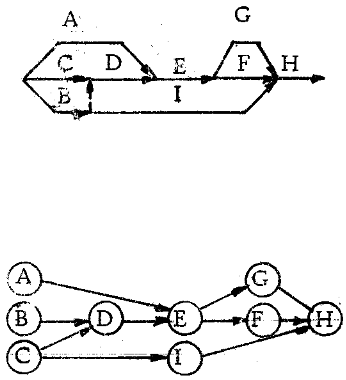


Figure 4.9. Patio example.

transducers which act as an AND fan-in (all prerequisites must be in before there can be an output from the transducer) and an AND fan-out (all postrequisites receive the same starting signal) are understood. This implies that all preceding states (prerequisite activities) have the same set of output transducers (/'s in their columns) and all succeeding states (postrequisite activities) have the same set of input transducers (/'s in their rows). When this fails, the transducer will have to be shown explicitly, thus, the birth of the ghost activity: "dummy." The network on the bottom of Figure 4.9 is our regular flowgraph showing explicitly both the states (activities in the circles) and the transducers (arrows).

Dummy examples. In order to find those dummy connections, it is only necessary to follow a simple three-step procedure: (the example corresponds to Figure 4.9)

1. Find all /'s in each row (or column); say a_{42} and a_{43}.

2. Compare all /'s in the columns (rows) corresponding to the /'s found in 1; say a_{42} and a_{92} in column 2 and a_{43} in column 3.

3. Those /'s that are contained in one column (row) but not contained in the other column (row), say a_{92}, corresponds a unique transducer, thus, the common transducer, a_{42}, shall now be explicitly shown as a dummy: Ɖ.

Obviously, the method can be used on all postrequisites (as shown above) or on all prerequisites (as shown within parentheses) and will yield the same dummies. Thus, the method is invariant whether systems tableau (Figure 4.9) or CPS tableau (Riggs and Inoue, 1966; Inoue, 1964) is used.

Several more examples are shown in Figure 4.10 for those interested.

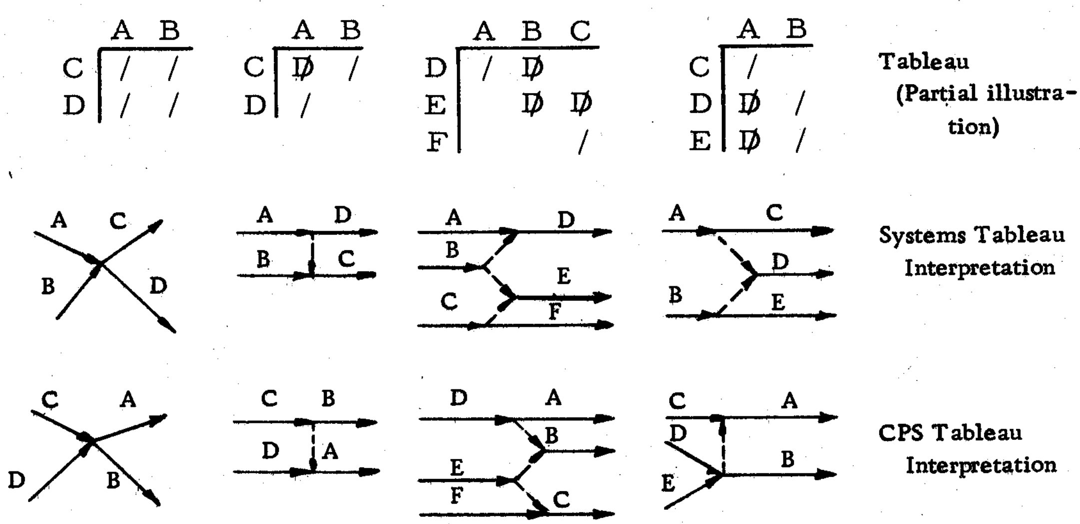


Figure 4.10. Examples of CPS dummies.

Mathematical Manipulations

Repeated Transformation

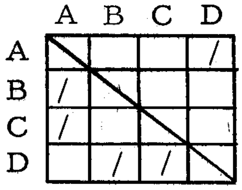
One-step transformation. A Tableau is basically a representation of all the one-step transformations that exist within a set of elements. Thus, we have the single step connection: state-transducer-state which indicates the single relation a_{ij} from state j to i. We can symbolically express the matrix of the Tableau as A.

Two-step transformation. Because of the eigen-line identification of states, it is visually possible to identify how two states are related through two transducers, by forming all possible two-step staircase patterns: state-transducer-state-transducer-state, or:

$$a_{ij}^{(2)} = \sum_{k=1}^{n} a_{ik}a_{kj}$$

We notice at once that this is equivalent of multiplying the matrix A by itself, or A^2.

Multiple-step transformation. This process may be repeated for any number of steps. The number of transducers involved in the staircase pattern determines the corresponding power of the multiplication of the matrix A. Examples are shown below in Figure 4.11 (see also page 87).



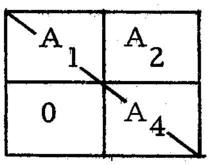
One-steps: A-B, A-C, C-D, B-C, D-A.
Two-steps: A-B-D, A-C-D, D-A-B, D-A-C, B-D-A, etc.
Three-steps: A-B-D-A, A-C-D-A, B-D-A-C, etc.

Figure 4. 11. Chained transducers.

Irreducibility

Definition 4. 1. An independent subset. The subset S of elements of the set composing the Tableau T, is said to be independent, if and only if the entry a_{ij} in T is zero for all $j \in S$ and i not in $S (i \notin S)$.

This means that the subset S may act as a source but not as a sink for any other element in the system. In the case of the economic trade example, we may say that these countries in S imports from each other but not from outside. Just as a sink element will have no entry in its column, an independent subset may be recognized as a partitioning of the Tableau in the form shown in Figure 4. 12.



A_4 is an independent subset (source).

A_1 is a sink subset.

(A_2 may be called a transducer subset.)

Figure 4. 12. Partitioning of a tableau.

Definition 4. 2. Irreducible Tableau. A tableau T is said to be irreducible, if there is no empty subset of T except T itself.

Definition 4.3. Irreducible group. A group G contained in tableau T is said to be irreducible if and only if it does not contain an independent subset, except the group itself.

This concept of reducibility becomes very important when we deal with the eigen value problems. We would like to show its relationship to Neumann inverse, and in passing, we shall also show a closely related concept of "clique" used in sociology.

Neumann inverse. By using the numerical value of 1 instead of /, we can numerically evaluate the number of paths that are present between any two states with m number of steps or transducers. For example, $a_{ij}^{(2)}$ will be the number of two-step chains available between j and i, and $a_{ij}^{(m)}$ will be the number of m-step chains available between j and i. We call such matrix A, a matrix of relation, and A^m, the m^{th} product of A.

The matrix which is the sum of all $A^{(m)}$ where $m = 1, \dots \infty$ is called Neumann Inverse (Inoue, 1964). In particular, the Neumann Inverse Q for a matrix A is given by:

$$Q = I + A + A^2 + A^3 + A^4 + \dots, A^n + \dots, = (I-A)^{-1}$$

hence the name Neumann "Inverse" (Inoue, 1964, p. 53).

Definition 4.4. A clique. A clique is any maximal collection of three or more elements having the property that any two elements in the group is related by a two-way transformation.

Theorem 4.1. Clique detection. The i^{th} element of a symmetric Tableau is a clique if and only if the three-step transformation of i to i is positive (there is an entry in the eigen-cell i).

Proof of Theorem 4.1. If $a_{ii}^{(3)}$ is positive, there must be at least one path a_{ij} a_{jk} a_{ki}. Thus, elements i, j, and k must form a clique. By the same token, if $a_{ii}^{(3)}$ is zero, there can be no group of more than two communicating with each other, hence no clique.

Theorem 4.2. Irreducibility. A Tableau T is irreducible if and only if the Tableau corresponding to Neumann Inverse will have no empty cell.

Proof of Theorem 4.2. Any entry in a Neumann Tableau will indicate either a direct or an indirect relationship between two elements. Having all cells filled means that there is a two-way relationship between all elements, or the entire Tableau is a clique. The only possible independent subset is the Tableau itself. On the other hand, if a Neumann Tableau would have an empty cell, there must be at least one transducer that is missing or at least one element that does not depend on one other state. By a suitable rearrangement of the tableau, the empty cell may be brought to the left bottom of the Tableau to present a structure similar to Figure 4.12.

Particular Forms of Interest

Asides from cliques and reducible forms, a structure tableau (with suitable rearrangement of rows and columns by reordering the elements in the set) may exhibit particular characteristics.

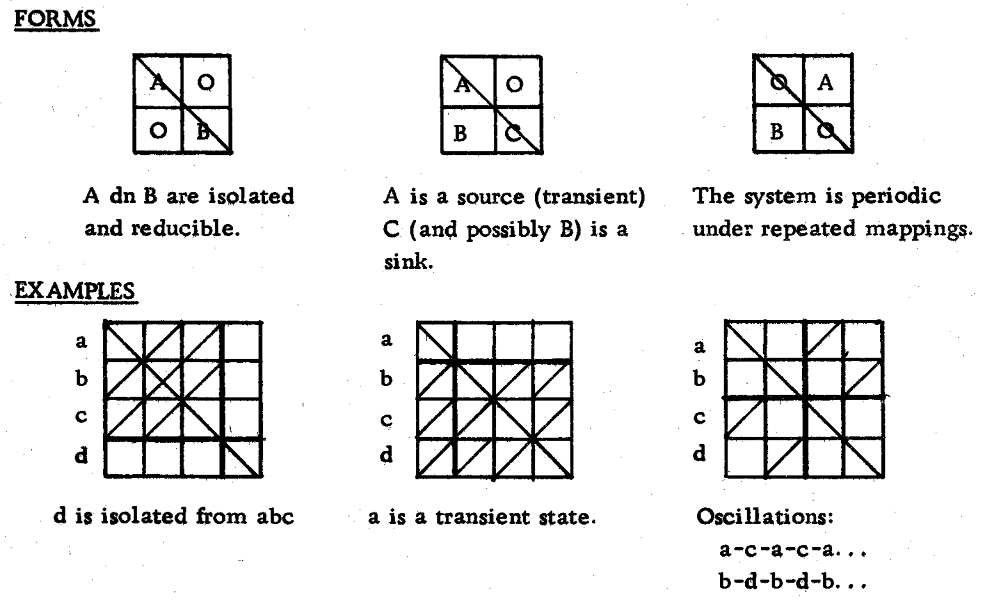


Figure 4. 13. Particular forms of a tableau.

Some of those are shown above in Figure 4. 13.

V. MODEL SYNTHESIS

> It is ironical in science that in order to understand we must throw away information. We cannot, at least at this level of our intellectual development, grapple with a high order of complexity. Consequently, we must simplify (Bellman and Kalaba, 1965, p. 5).

Orthogonalization

Quantitative Models

The use of Tableau method is obviously limited unless we are able to evaluate numerical quantities and arrive at quantitative as well as qualitative models of actual systems. Acquiring data and rendering them suitable to available analytic methods are, no doubt, the two most difficult steps in the Systems Study. Having a structural model such as those discussed in Chapter IV may be helpful but is a far cry from having synthesized a model for further quantitative studies.

In this chapter, we would first like to discuss orthogonalization and its practical significance away from the usual mathematical, physical, or engineering interpretations.

Next, we would like to discuss some of the assumptions that could be made in finding discrete parameters to use for models and to illustrate some queueing theory applications of Tableau Analysis.

Finally, we would like to present a simple model of a communication model synthesized using an elementary Hamming's code, and briefly discuss the problem of pattern recognition using a State Tableau.

Tableau as an Orthognal System Representation

Definition 5.1. An empty Systems Tableau of dimension n x n is a representation of a system of n degrees of freedom, each eigen cell representing an independent state in which the system may find itself. An empty Systems Tableau is defined as a square grid of n columns and n rows with no entry in any cell excepting the eigen line.

An empty Systems Tableau of dimension 4 x 4 is shown in Figure 5.1 together with its flowgraph representation. The four states were named x_1, x_2, x_3, and x_4.

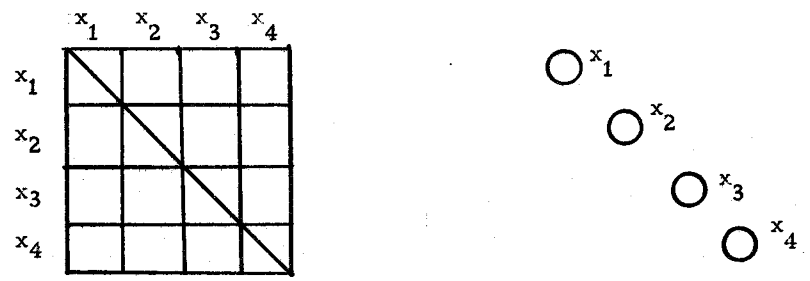


Figure 5.1. A system with four-degree of freedom.

Definition 5.2. An orthonormal tableau of dimension n x n is defined as a systems tableau of n x n with each eigen cell corresponding to an orthonormal state.

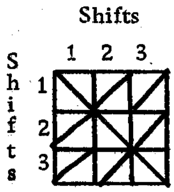
Theorem 5.1. From a given system of infinitely many states $x_1, x_2, x_3, \ldots$, and r of which are linearly dependent for arbitrary r, an orthonormal tableau with states $y_1, y_2, \ldots y_r$ may be obtained by taking y_j as a suitable linear combination of $x_1, \ldots x_j$.

Proof of Theorem 5.1. This is a restatement of Schmidt's orthogonalization process. Hadley (1961, p. 47) has an example of vector orthogonalization. Courant and Hilbert (1953, p. 50) have a discussion of Orthogonalization of Functions. In essence, the procedure is as follows.

First, choose x_1, any of the given state. Normalize it by finding the "length" or the absolute value of the state based on some arbitrary measure (an eigenvalue, for example) and dividing x_1 by it: $y_1 = x_1/|x_1|$. y_2 is then found by making x_2 independent of y_1: $y_2 = [x_2 - (y_1x_2)y_1]/|x_2 - (y_1x_2)y_1|$ where (y_1x_2) is an inner product of the two states. (We may consider this as a dot product of the two state vectors, or the projection of a state into the other. In particular, we define (x_1x_1) as the length of the state vector x_1.) This process is continued by adding a new state each time and defining an orthonormal state. When we have covered r of them, the rest will be linear combinations of the first r orthonormal basis.

Example 5.1. Let us illustrate this concept with a very simple example. Let us suppose that a newspaper company has three shifts of ten working hours each on two newspaper editions (e.g. morning and evening). A worker can be in any one of the shifts 1, 2, or 3. Shift 1 works exclusively on edition A of the morning while 2 and 3 will work part-time on edition A and part-time on the evening edition B.

A possible tableau may look like Figure 5.2.



/ indicates the possible relationships between the states. For example, a_{12} has a / showing that a man in shift 1 may work on edition A with a man in shift 2.

Figure 5.2. Non-orthogonal example.

Now let us assume that we wish to find a set of orthonormal bases to form an orthonormal tableau. We select shift 1 to find the first basis. Since shift 1 works for ten hours, in this simple example, we may take $y_1 = 10x_1/|10| = x_1$. If the second shift works two hours on A and eight hours on B, then,

$$y_2 = [10x_2 - (y_1x_2)y_1]/|10x_2 - (y_1x_2)y_1|$$

But (y_1x_2) is essentially the time of x_2 spent on y_1 (edition A) or two hours. Thus,

$$y_2 = (10x_2 - 2y_1)/8$$

In our particular case, y_1 happens to represent the state of working for edition A, and y_2 that of working for B.

y_1 = A

y_2 = b since $10x_2 - 2A = 8$ hours spent on B, and $8/8 = 1$.

The third shift x_3 can now be represented in terms of the two orthogonal bases y_1 and y_2, or A and B. Assuming that a worker in shift 3 devotes eight hours to A and two hours to B, the obvious result is obtained:

$$x_3 = 10 = 8A + 2B$$

Thus from three states 1, 2, and 3, we have obtained the two orthonormal states A and B. If all three states were independent from the start, we would have obtained three orthonormal states. The inner products for the orthonormal states yield: (AA) = 1, (AB) = 0, (BA) = 0, (BB) = 1.

The example was made overly simple to illustrate only the concept. Readers can easily make the example more elaborate and realistic. For instance, we may say that the second shift spends 85 percent of the first two hours on edition A, [or even better, attributes a statistical correlation yielding some sophisticated (y_1x_2), etc.].

The orthonormal tableau is shown in Figure 5. 3 using the states A and B instead of shifts 1, 2, and 3.

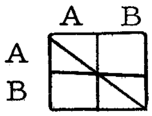


Figure 5.3. Orthonormalized systems.

Definition 5.3. A transposed tableau is a tableau in which each cell a_{ij} has been replaced by the cell a_{ji}. In an ordinary tableau the input or exogenous flows are horizontal while the output or endogenous flows are vertical. In a transposed tableau the corresponding flows would be shifted 90°.

Note. In many instances such as CPS (see Chapters IV and VIII), it may become more profitable to use a transposed tableau rather than the primary tableau. The operations described thus far are applicable to either provided that the terms "column" and "row" are interpreted as "primary input (exogenous) direction" and "transverse or transposed output (endogenous) direction."

Definition 5.4. A state that is an independent subset (Def. 4.1) is called a source, while a state corresponding to an independent subset in its transposed tableau is called a sink.

Probabilistic Interpretation

Sample space. The tableau representation can be interpreted in a probabilistic sense without ambiguity. A tableau is essentially a sample space description, where:

> The sample description space of a random phenomenon usually denoted by the letter S, is the space of descriptions of all possible outcomes of the phenomenon (Parzan, 1960).

The definition may be extended to include impossible outcomes by assigning the probability value of zero. Thus, a simple example of a toss of a die may be represented by Figure 5.4. An even can therefore be considered as a "state."

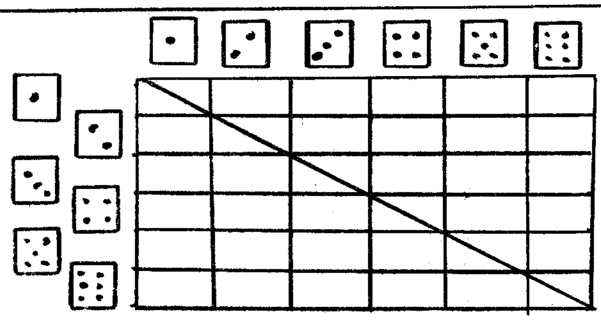


Figure 5. 4. A toss of a die tableau.

Events. Some times a subset of the sample space S is called an event. The Tableau's outcomes may be divided into events such as: all even numbers A(2, 4, 6); all primes B(2, 3, 5)[10]; etc. An event may contain other events or be contained in other events. For example, a single outcome "2" (say event C) may be a subset of both the events B and A, as well as of S. An example of this will be in our Figure 5.18 where an event "Subject" included either a noun or a pronoun, etc.

---

[10]A mathematician does not consider 1 to be a prime number. Otherwise all numbers will be divisible by 1, a prime number, and no other number would fit the definition of being a prime number (not being divisible by any other integer).

Disjoint Events

> Any two events, E and F, that cannot occur simultaneously, so that their intersection EF is the impossible event, are said to be mutually exclusive (or disjoint) (Parzan, 1960).

A very good example of this may be seen from our single-toss-of-a-die description. We have defined each particular outcome to be the state of the die showing one of its six faces in a horizontal plane. All other faces are perpendicular to the particular face, except of course for the face hidden under. In a sense, this presents a pictorial representation of our "Schmidt's Orthogonalization" we have illustrated in Example 5. 1. The transient event, "standing on its edge or on its corner," is eliminated by giving a little "perturbation" until the die will come to rest in one of the defined outcomes. Alternately, we could have accepted the transient state and decomposed it in terms of the orthogonal basis. For example, standing on an edge could have been broken down into two components corresponding to the two adjacent faces, say 2 and 3. Figure 5. 5 may clarify this concept.

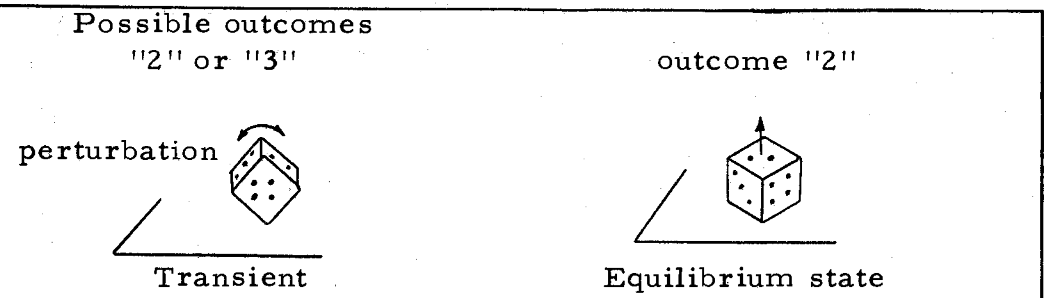


Figure 5. 5. A simple example of orthogonal relationships.

> In so far as probability theory is the study of mathematical models of random phenomena, it cannot give rules for the construction of sample description spaces. Rather the sample description space of a random phenomenon is one of the undefined concepts with which the mathematical theory begins. The considerations by which one chooses the correct sample description space to describe a random phenomenon are a part of the art of applying the mathematical theory of probability to the study of the real world (Parzan, 1960, p. 11).

If we remember that the model we like to build of the single toss of a die is a probability model rather than a dynamic model, it becomes clear that the rules to be used in orthogonalizing the transient state must not be a geometric consideration involving the cosines of the angles of the planes of the die and the table on which the die is found, but rather a probabilistic consideration based on their fundamental axioms. In other words, the transient state of in-equilibrium should not be decomposed into $(2)^{-1/2}$state 2 + $(2)^{-1/2}$ state 3, but instead into Prob (state 2) and prob (state 3). If the die has an equal chance of moving into state "2" as well as the state "3" then the probability of transition should be one-half in either case.

A Tableau representing a system description in terms of probabilities of transitions will be called a Probability Tableau.

<u>Probability Tableau.</u> A probability Tableau is defined as a Tableau that meets the three axioms of the probability theory:

> Given a random situation, which is described by a sample description space S, probability is a function P(.) that to every event E assigns a nonnegative real number denoted by P(E) and called the probability of

the event E. The probability function must satisfy three axioms:

Axiom 1. $P(E) \geq 0$ for every event E.

Axiom 2. $P(S) = 1$ for the certain event S.

Axiom 3. $P(E \cup F) = P(E) + P(F)$, if $EF = \emptyset$, or in words, the probability of the union of two mutually exclusive events is the sum of their probabilities (Parzan, 1960).

The first axiom may be interpreted as the fan-in rule. All preceding states should be related to their next states by a non-negative transmittance. In other words, no cell may contain a negative number.

The second axiom is the fan-out rule: each column must add up to 1 since the tableau is a description of all possible states in which the system may find itself.

The third axiom is the transfer condition that would have been satisfied if the tableau were orthogonalized beforehand.

A simple example of a Markov Process depicting the transfer of customers between three firms A, B, and C is shown below in Figure 5. 6. (Note that a Tableau is a transpose of the usual Markov or Stochastic matrix.)

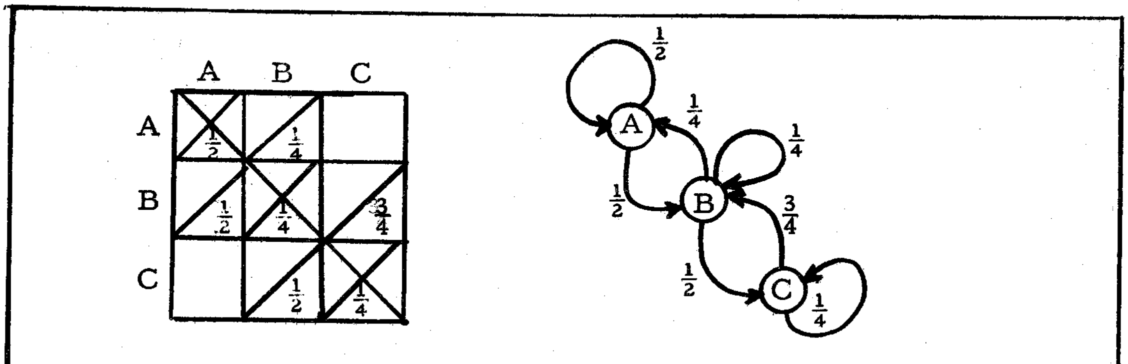


Figure 5. 6. A probability tableau example.

Queueing Models

Parametric Representations

Big O, little o notation. Another very legitimate method would be to use either Moment Generating Function, or parameters of the distribution that characterize the random variable. In such cases, it may be necessary to start with a discrete time series and work toward a continuous case to obtain the expression needed to characterize the state.

This approach (which could be used in deriving probability distributions, for example), was formalized in the so-called "Big O, little o" notations by Professor Ronald W. Wolff of the Department of Industrial Engineering at the University of California at Berkeley (Wolff, 1966).

According to his notations:

o(t) denotes any function of time t such that:

$$\lim_{t \to o} \frac{o(t)}{t} = O \qquad \text{and,}$$

O(t) denotes any function of time t such that:

$$\lim_{t \to o} \frac{O(t)}{t} = \text{finite.}$$

Then, using these notations, we can develop a peculiar short-hand algebra:

$$o(t) \pm o(t) = o(t)$$

$$o(t) \pm O(t) = O(t)$$

$$o(t)O(t) = o(t)$$

$$O(t) \pm O(t) = O(t)$$

$$ko(t) = o(t) \text{ for a finite k, a scalar.}$$

$$o[O(t)] = o(t) = O[o(t)]$$

$$O[O(t)] = O(t)$$

Even though these results have been used often in the past, and are intuitively obvious, no such formalization has been attempted previously (to the best of the author's knowledge).

Derivation of a Poisson distribution.[11] As an example to show how this notation may be used, and also to remind us of the origin of Poisson distribution, let us follow its derivation from the binomial (Bernoulli) distribution.

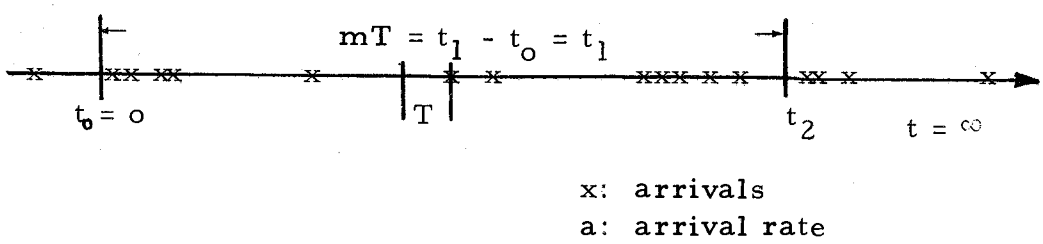


Figure 5.7. Random arrivals.

---

[11]The Poisson probability law was first published in 1837 by the French mathematician S. D. Poisson in his book "Recherches sur la probabilité des jugements en matière criminelle et en matière civile," (Research on the probability of judgements on criminal and civil matters). Thus, the early applications were to such phenomena as the suicides of women and children and deaths of Prussian soldiers from kicks of horses in Prussian Army (Parzan, 1960, p. 225).

We take a small segment of time, T, such that $mT = t_1 - t_o = t_1$, and assume that the probability for the arrival of one customer in that segment T is given by:

$$T + o(T) = \text{Probability of arrival of one customer.}$$

Then, $1 - [T + o(T)] = 1 - aT + o(T)$ (from the fifth relation by setting k = -), is the probability that more or less than one customer will show up. Since the probability for more than or equal to two customers showing up is also o(T), we have:

$$\text{Prob(no customer)} = \text{Prob(0 or more than 2)}$$

$$- \text{Prob(more than 2)}$$

$$= - aT + o(T) - o(T) = 1 - aT + o(T)$$

from the first relation.

Thus, applying a binomial distribution to the probability of having r number of customers in interval mT, we obtain:

$$\text{Prob}(r \text{ in } mT) = (\tfrac{m}{r})\,[\text{prob(no cust)}]^{m-r}[\text{prob(1 cust)}]^{r}$$

$$\text{Prob}(r \text{ in } mT) = \lim_{\substack{T \to o \\ m \to \infty}} \binom{m}{r}[1 - aT + o(T)]^{m-r}[aT = o(T)]^{r}$$

$$= \lim_{\substack{T \to o \\ m \to \infty}} \frac{m!}{r!(m-r)!}\,(1 - aT)^{m-r}(aT)^{r} \text{ by neglecting o(T)'s}$$

And by introducing $\frac{t_1}{m} = T$, we obtain:

$$= \frac{1}{r!} \lim_{m \to \infty} \frac{m!}{(m-r)!}\,(1 - a\frac{t_1}{m})^{m-r} a^{r} \frac{t_1{}^{r}}{m^{r}}$$

$$= \frac{a^{r} t_1{}^{r}}{r!} \lim_{m \to \infty} \frac{m!}{m^{r}(m-r)} \lim_{m \to \infty} (1 - \frac{at_1}{m})^{m-r}$$

Now, we note that:

$$\frac{m!}{m^r(m-r)!} = \frac{m(m-1)(m-2)\ldots.(m-r+1)(m-r)(m-r-1)\ldots..1}{m \quad m \quad m \ldots. \quad m \quad (m-r)(m-r-1)\ldots..1}$$

$$= \frac{m}{m}(1-\frac{1}{m})(1-\frac{2}{m})\ldots\ldots(1-\frac{r+1}{m})$$

$$= 1[1-o(1/m)][1-o(1/m)]\ldots..[1-o(1/m)]$$

and therefore for $(1/m) \to O$ or $m \to \infty$, we obtain:

$$\lim_{m\to\infty}\left(\frac{m!}{m^r(m-r)!}\right) = 1$$

Also:

$$\lim_{m\to\infty}(1-\frac{at_1}{m})^{m-r} = e^{-at_1}$$

Hence,

$\text{Prob}(r \text{ arrival in } t_1) = \dfrac{e^{-at_1}(at_1)^r}{r!}$ which is the Poisson distribution.

Models

Birth and death process. A Tableau is well suited for modelling a queueing process, and especially so when the interrelationships are complex. In order to illustrate how a Tableau can be used, we shall borrow an example presented by Wolff.

Example Problem 5. 2.

Consider a queue with two exponential stations in tandem, each with rate u, and Poisson arrivals at rate v, operating with the following rules:

1. No queue is allowed in front of either station.
2. Both stations may operate simultaneously except that if an item is station 1 is completed while station 2, is busy, it must wait in station 1, blocking station 1 from working on a new item

until station 2, if free. Arrivals to station 1, while it is busy or blocked, are lost (Wolff, 1966).

In this problem, we are mainly interested in the proportion of arrivals lost and the mean number of busy (unblocked) stations.

Our first attempt in analyzing such a problem would probably result in some analysis in phase space. A diagram such as the one shown in Figure 5. 8 may be helpful in understanding the problem, but utterly hopeless in providing any solution. A description of the problem using that diagram will be almost as wordy as the problem statement itself.

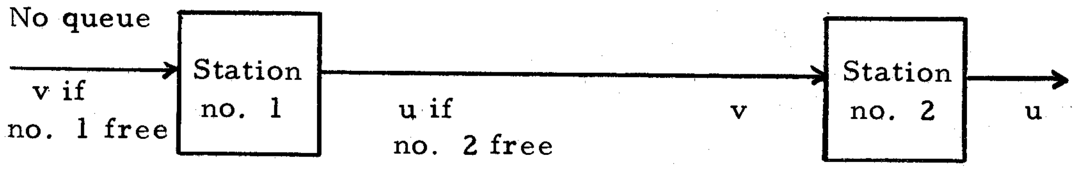


Figure 5. 8. Phase space description of the sample queueing problem.

The main difficulty occurs from the fact that the phase space does not allow us to identify disjoint events since the same physical objects may be in different states in different time. On the other hand, we notice that a description of the system in terms of possible states has the effect of orthogonalizing the systems description. The system can be in one state only at any particular instant of time, and all states can be expressed in orthogonal terms by a suitable treatment of transient states either through some perturbation or by giving recognition of such transition states as legitimate states in their own rights.

Using a shorthand notation (a, b) we can identify the five possible states of the system: (0, 0) no station is occupied; (1, 0), only the first station busy; (0, 1), only the second busy; (1, 1) both busy; and (B, 1) when the first one is blocked by the work in the second. Figure 5. 9 shows its Tableau.

	(00)	(10)	(01)	(11)	(B1)
(00)			u		
(10)	v			u	
(01)		u			u
(11)			v		
(B1)				u	

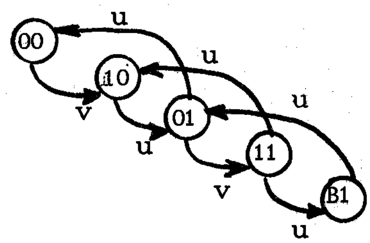
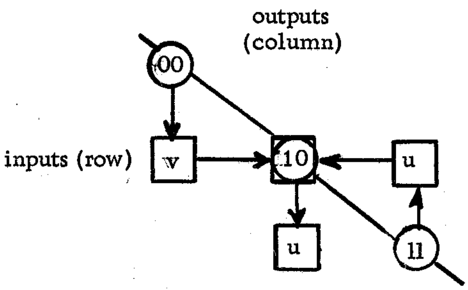


Figure 5. 9 . a. State tableau for queueing example.



$$vp_{00} + up_{11} = up_{10}$$

Figure 5. 9 . b. An example of a birth and death equation.

After the Tableau has been built, the analysis becomes purely mechanical: we merely need to balance input and output at each eigen-cell (state). Thus, by equating columns (input) to rows (output), we obtain the birth and death balance equations as follows:

State (phase)	Rows (Input)		Columns (Output)		
00	up_{01}	=	vp_{00}	or	$p_{01} = \frac{v}{u}p_{00} = QP$
10	$up_{11} + vp_{00}$	=	up_{10}	or	$p_{10} = (\frac{Q^2}{2} + Q)P$
01	$up_{B1} + up_{10}$	=	$(v+u)p_{01}$		(redundant)
11	vp_{01}	=	$2up_{11}$	or	$p_{11} = \frac{Q^2}{2}P$
B1	up_{11}	=	up_{B1}	or	$p_{B1} = \frac{Q^2}{2}P$

where p_{ij} = Probability (state is a = i, b = j)

$P = P_{00}$ = Prob(vacant system)

$Q = \frac{v}{u}$.

The total probability for the system is:

$$T = p_{00} + p_{01} + p_{10} + p_{11} + p_{B1} = (\tfrac{3}{2}Q^2 + 2Q + 1)P$$

The probability that the system cannot accept a new input (i.e. station a blocked or occupied) is:

$$p_{10} + p_{11} + p_{B1} = (\tfrac{3}{2}Q^2 + Q)P$$

And the proportion of arrivals lost is: $\dfrac{3Q^2 + 2Q}{3Q^2 + 4Q + 2}$.

The mean number of busy station is:

$$B(Q) = \frac{p_{01} + p_{10} + p_{B1} + 2p_{11}}{T} = \frac{4Q^2 + 4Q}{3Q^2 + 4Q + 2}$$

and the average number of customers in the system will be:

$$L = B(Q) + \frac{p_{B1}}{T} = \frac{5Q^2 + 4Q}{3Q^2 + 4Q + 2} .$$

Example Problem 5.3. Erlang Service. Before leaving this topic of Queueing models, let us illustrate one more example of Tableau using a simple Erlang model. We can conceptually divide the system into five parts, each corresponding to 20 percent of the work. Each customer must go through k = 5 stages of services at the rate of ku. The phase-space model and the state-space Tableau are shown in Figure 5.10 below (adapted from Wolff, 1966).

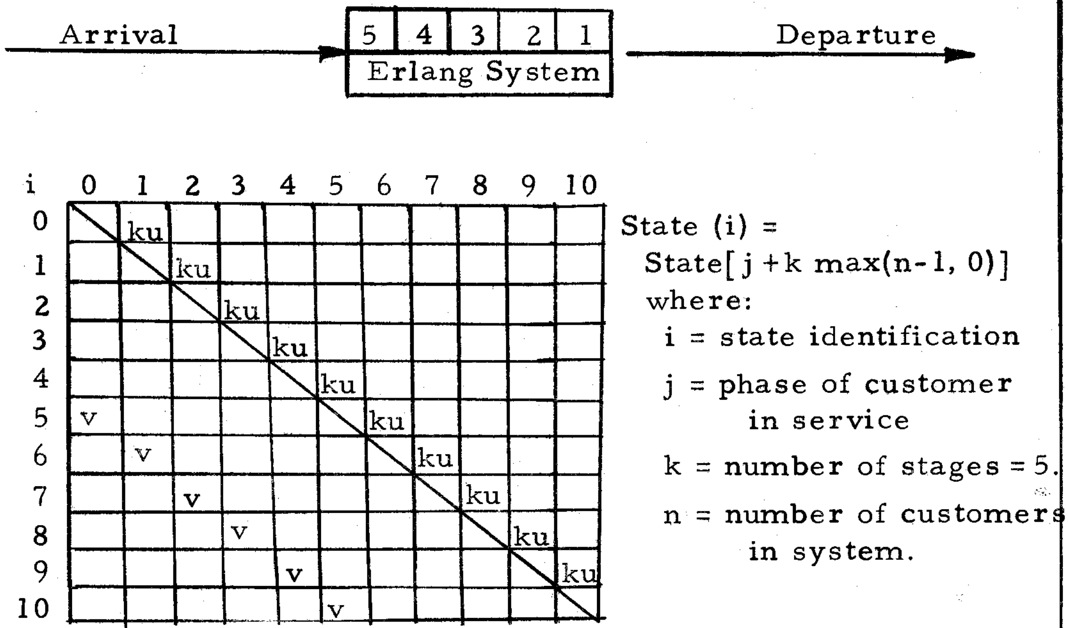


Figure 5.10. Erlang model.

This is also an example of how a discrete approach can be used to simulate a more continuous process. By increasing the number of k, the frequency distribution can be made even more accurate.

Communication Systems

Coding, Channel, and Decoding

In Chapter I (Figure 1. 5) we have briefly touched upon the subject of a communication system. We would now wish to examine a simple example that will illustrate the applicability of Tableau in representing and constructing such a system.

Example Problem 5. 4. Let us suppose that we are interested in sending a simple binary message over a noisy channel. The message "yes"(1) or "no" (0) will have to be decoded at the receiving end.

Definition 5. 5. A binary operation AND will be symbolized by a dot (·) and its operation defined by the truth table below:

A	B	A· B
0	0	0
0	1	0
1	0	0
1	1	1

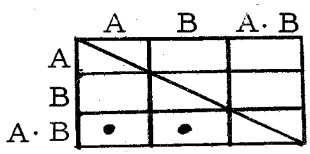


Figure 5. 11. AND operation.

Definition 5. 6. A binary operation OR will be symbolized by a plus (+) and its operation defined by the truth table below:

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

	A	B	A+B
A			
B			
A+B	+	+	

Figure 5.12. OR operation.

Problem. We wish to synthesize an encoder that will translate the message x into codes a_1, a_2, and a_3; transmit it over the channel; and decode the received signal b_1, b_2, and b_3 into the message y.

Figure 5.13 illustrates the black box representation of this problem.

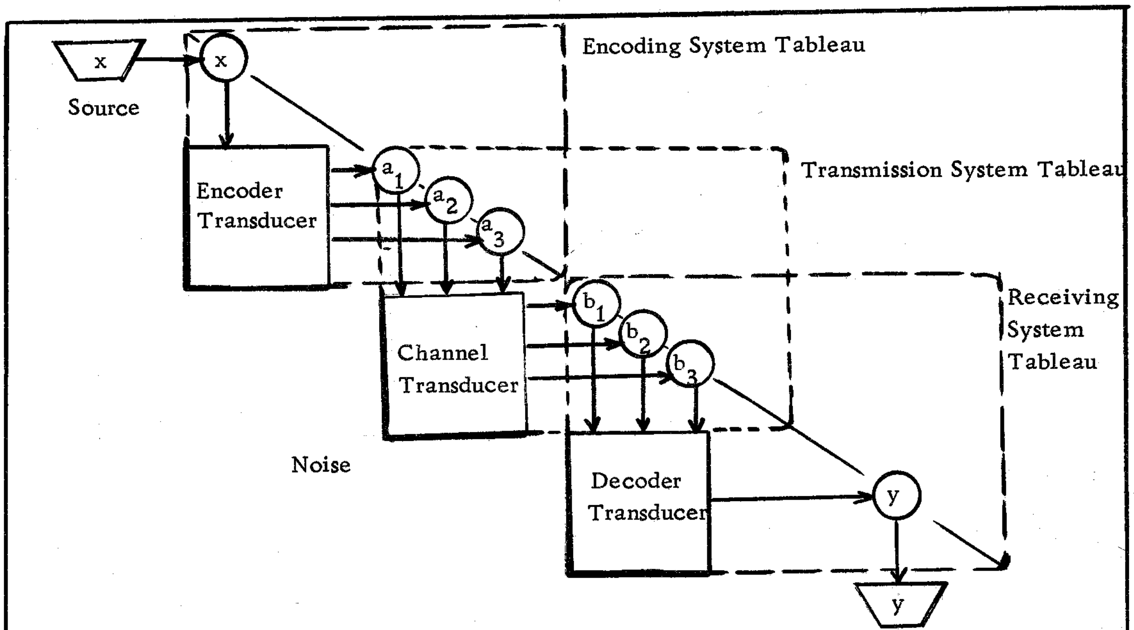


Figure 5. 13. Binary code transmission.

In Chapter I we have already shown how "bits" are calculated. In our problem the message is a simple binary information (say

H = 1 bit/second) while the channel has the possibility of transmitting 000, 001, 010, 011, 100, 101, 110, 111 or eight combinations of a_1, a_2, and a_3. Thus, C = 3 bits/second = channel capacity (no noise).

Self-correcting code. Hamming has developed a single-error-correcting code that will decode the received signal in such a way that a single error of 0 being received as a 1 (similar to a "False-alarm" or the error of type II) or that of receiving a 1 as a 0 (a "Miss" or the error of type I) can be corrected automatically. The mechanism becomes clear for our simple case by referring to Figure 5.14. Of course, no statement of the quality of the detector itself (the device that decides whether 0 or 1 has been received) has been made. Figure 5.14 shows that if "1" has been coded as 111 and "0" as 000 in $a_1a_2a_3$, the received signal $b_1b_2b_3$ can still yield the correct message even under a distortion causing a single miss or false-alarm, if an homomorphic transformation would map 110, 101, 011, and 111 into y = 1 and 001, 010, 100, and 000 into y = 0. In other words, we are utilizing the fact that the only way to transfer from 000 to 111 in Figure 5.14 is by passing through three edges (three mistakes) and there is no shorter path. For more advanced types of Hamming's codes, readers are referred to Chu (1962, p. 84), Ash (1965, Chapter 4), Arbib (1964, p. 77), etc.

Encoder and decoder design. Based on the Hamming code the encoder tableau and the decoder Tableau have been designed as shown

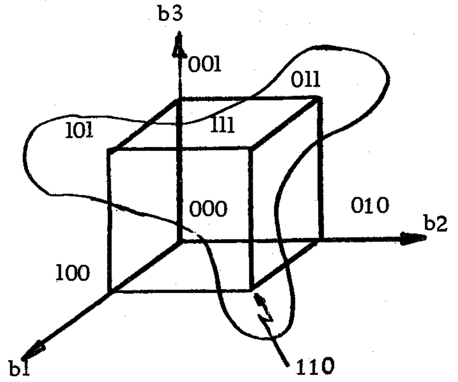


Figure 5. 14. Hammings' code.

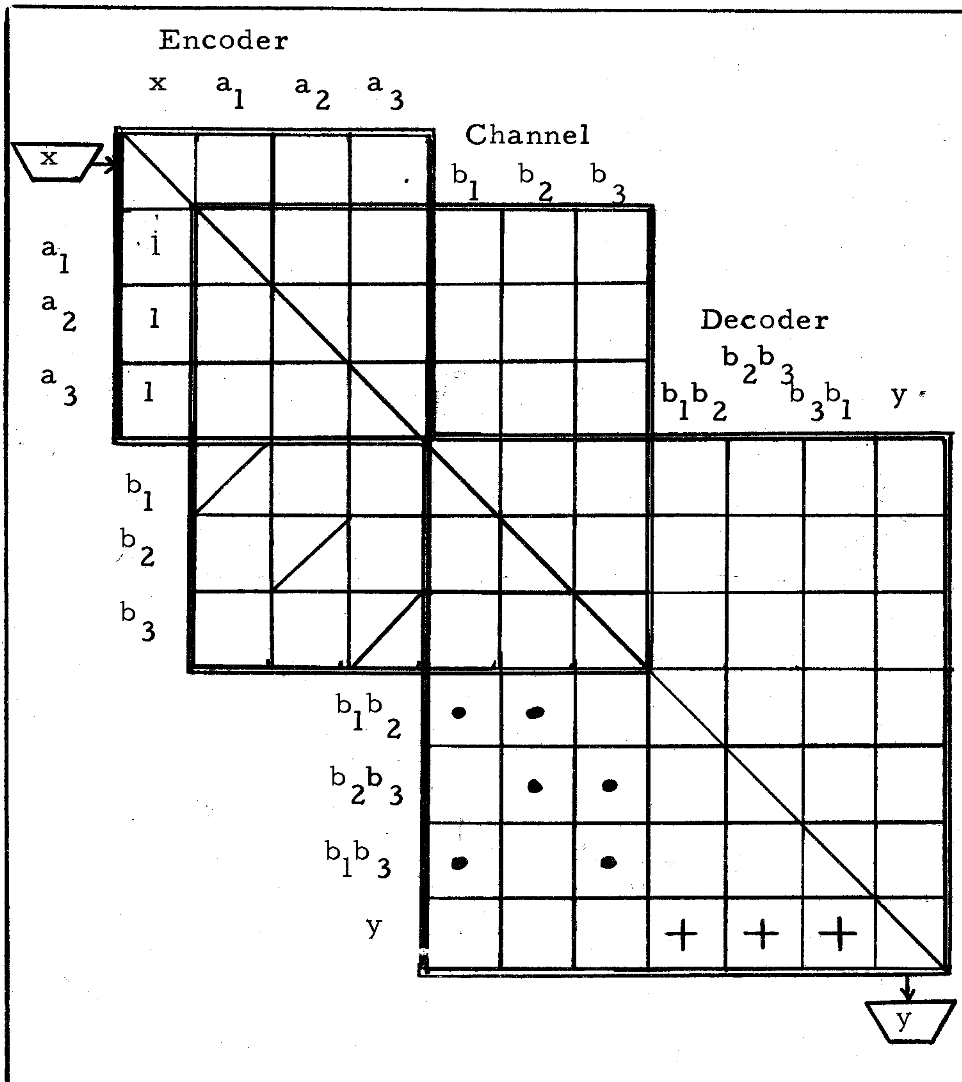


Figure 5. 15. Tableau realization of Hammings' code.

on Figure 3.15. The encoder is essentially an AND fan-out distribution. If the initial signal is strong enough with respect to the input impedances of the channels, it can be directly connected to them. If not, an amplifier may have to be used.

The decoder is composed of three AND fan-ins and one OR fan-in. It is possible to have used a three-input majority voting circuit that would give 1 if two or more of the three inputs were 1 and 0 otherwise. In the terminology of Nilsson (1966, p. 99) this would be a "committee machine" used in a learning machine.

Figure 5.16 reveals the striking similarity between our Tableau and an equivalent electrical switching circuit that could (at least theoretically) perform an actual coding and decoding task according to the Hamming's code.

Integrated Tableau. The three tableaux corresponding to the encoder, channel and decoder may be combined into a single tableau as shown in Figure 5.17. However, in order to simplify this tableau further, it becomes necessary to understand the nature of our channel. So far the only assumption we have made is the existence of some correlation between the a's and the b's. We used the usual / mark to indicate this relationship without having to specify it further: e.g. a_1/b_1, a_2/b_2, ... etc.

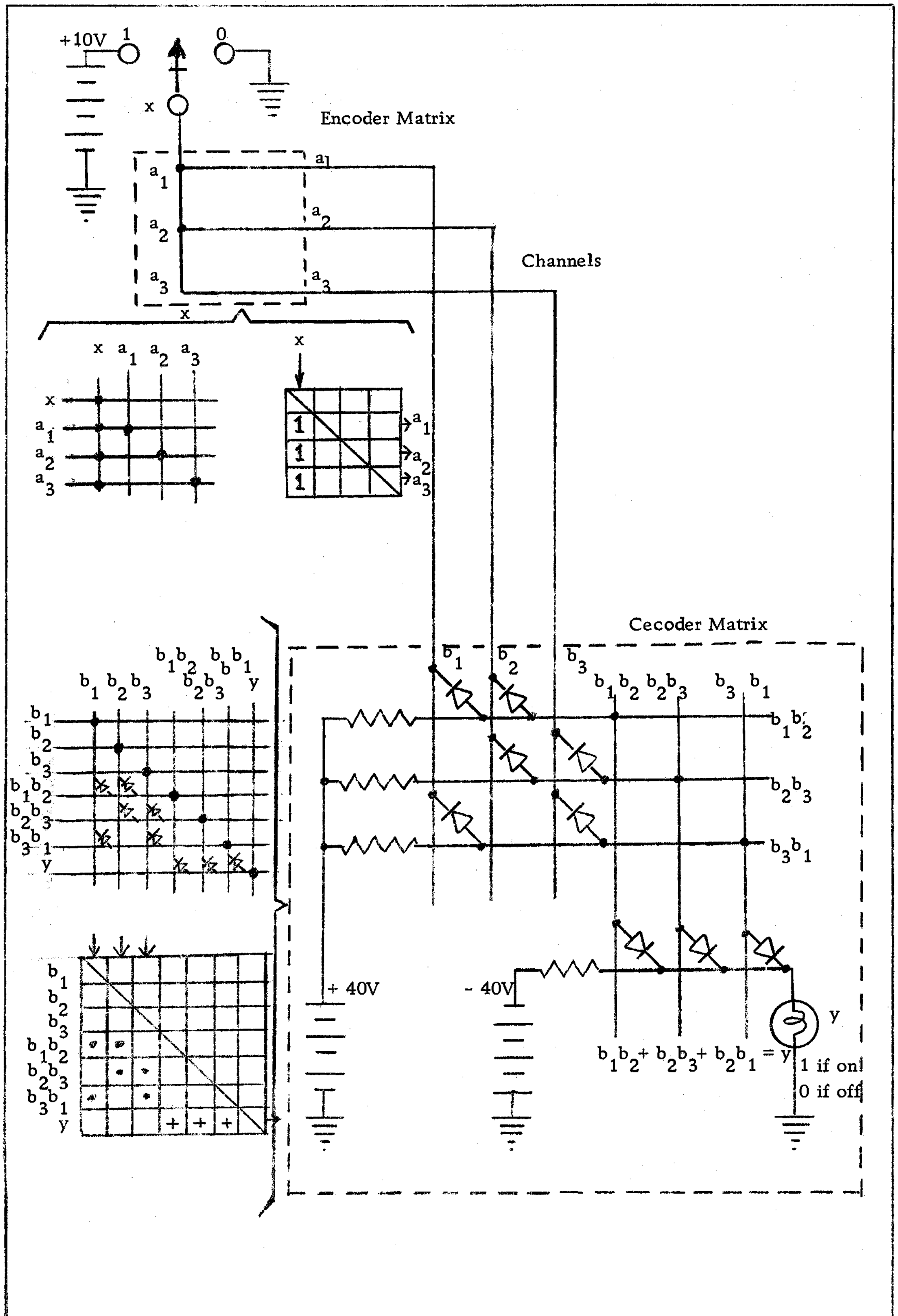


Figure 5. 16. Diode switching circuit designed from the tableau.

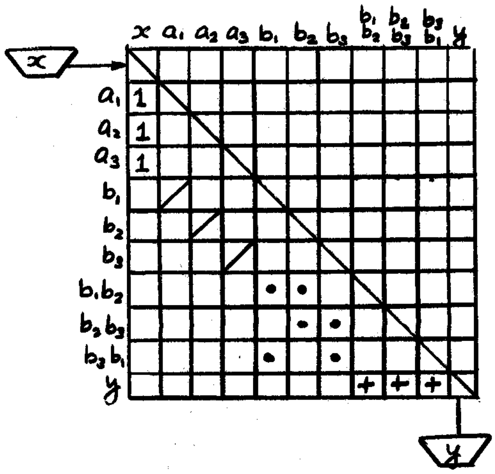


Figure 5.17. Integrated Tableau.

State Tableau Concept

Swift, a forerunner of Shannon?

Swift. . . described a machine invented by a professor of Laputa (Gulliver's Travels, Part III, Chapter 5). This contrivance was a 20 ft. square frame containing hundreds of small cubes linked together by wires. On each face of every cube was written a Laputan word. By turning a crank, the cubes were rotated to produce random combinations of faces. Whenever a few words happened to come together and make sense, they were copied down; then from these broken phrases erudite treatises were composed. In this manner, Swift explained, 'the most ignorant person at a reasonable charge, and with a little bodily labour, may write books in philosophy, poetry, politics, law, mathematics, and theology, without the least assistance from genius of study' (Gardner, 1958, p. 2).

For the benefit of our deserving readers who might be tired of reading the writing by the author, let us see what our tableau can do to remedy the situation and introduce new styling.

Shannon, in his famous treatise on Communication Theory (1949) introduced Markov Process as a model for information sources. His model regarded it as the zero-order approximation of characters (independent symbols and random choice with uniform probability: e. g. XFOML RXKHRJ FFJUJ, etc.); first-order approximation of characters (independent but probability based on English: e. g. OCRO HLI RGWR, etc., see Table 5. 1); second-order approximation of characters (English probabilities extended to interrelationships between each group of three or less characters: e. g. ON IE ANTSOUTINYS ARE. . .); to the second-order word approximation (English words are picked according to the probabilities of association: THE READ AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED).

In this last example, Shannon has provided correct transition probabilities by looking up a word at random from a page of a book, writing it down, scanning the next page for the same word, writing down the word next to it, etc... In this manner, he has completely dispensed himself of the random number

Table 5.1. Information theory data.

Composition of English Language

Letter	Frequency	Morse Code
Space	17.0%	
E	8.9%	.
T	7.1%	—
A	5.8%	. —
O	5.0%	— — —
Others	56.2%	

$\log_2 N = 3.322 \log_{10} N$

$+ \log_2(n) = \text{bit}(1/n)$

n	0	1	2	3	4	5	6	7	8	9
0.0		0.000	1.000	1.585	2.000	2.322	2.585	2.807	3.000	3.167
10.0	3.322	3.460	3.585	3.700	3.807	3.907	4.000	4.098	4.170	4.248
20.0	4.322	4.387	4.460	4.524	4.585	4.644	4.701	4.755	4.807	4.858
30.0	4.907	4.955	5.000	5.043	5.098	5.129	5.167	5.210	5.248	5.286
40.0	5.322	5.358	5.387	5.426	5.460	5.488	5.524	5.555	5.585	5.615
50.0	5.644	5.672	5.701	5.728	5.755	5.782	5.807	5.834	5.858	5.883
60.0	5.907	5.931	5.955	5.974	6.000	6.022	6.043	6.066	6.098	6.109
70.0	6.129	6.150	6.167	6.190	6.210	6.229	6.248	6.266	6.286	6.304
80.0	6.322	6.334	6.358	6.375	6.387	6.420	6.426	6.443	6.460	6.476
90.0	6.488	6.508	6.524	6.539	6.555	6.570	6.585	6.600	6.615	6.628
100.0	6.644	6.658	6.672	6.687	6.701	6.714	6.728	6.743	6.755	6.768

$- \log_2 p = \text{bit}(p)$

p	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0		6.644	5.644	5.059	4.644	4.322	4.059	3.837	3.644	3.474
0.1	3.322	3.p84	3.059	2.943	2.837	2.737	2.644	2.557	2.474	2.396
0.2	2.322	2.252	2.184	2.120	2.059	2.000	1.943	1.889	1.837	1.786
0.3	1.737	1.690	1.643	1.599	1.556	1.515	1.474	1.434	1.396	1.358
0.4	1.322	1.286	1.252	1.218	1.184	1.152	1.120	1.089	1.059	1.029
0.5	1.000	0.972	0.943	0.916	0.889	0.862	0.837	0.811	0.786	0.761
0.6	0.737	0.713	0.690	0.667	0.644	0.621	0.599	0.578	0.556	0.535
0.7	0.515	0.494	0.474	0.454	0.434	0.415	0.395	0.377	0.359	0.340
0.8	0.322	0.304	0.286	0.269	0.252	0.234	0.218	0.201	0.184	0.168
0.9	0.152	0.136	0.120	0.105	0.089	0.074	0.059	0.044	0.029	0.015

$-p \log_2 p = p\, \text{bit}(p) = H(p)$

p	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0		0.066	0.112	0.152	0.186	0.216	0.244	0.269	0.292	0.313
0.1	0.332	0.350	0.367	0.383	0.397	0.411	0.423	0.435	0.445	0.455
0.2	0.464	0.473	0.481	0.488	0.494	0.500	0.505	0.510	0.514	0.518
0.3	0.521	0.524	0.526	0.528	0.529	0.530	0.531	0.531	0.530	0.530
0.4	0.529	0.528	0.526	0.524	0.521	0.518	0.515	0.512	0.508	0.504
0.5	0.500	0.495	0.491	0.485	0.480	0.474	0.468	0.462	0.456	0.449
0.6	0.442	0.435	0.428	0.420	0.412	0.404	0.396	0.387	0.378	0.369
0.7	0.360	0.351	0.341	0.331	0.321	0.311	0.301	0.290	0.280	0.269
0.8	0.258	0.246	0.235	0.223	0.211	0.199	0.187	0.175	0.162	0.150
0.9	0.137	0.124	0.111	0.097	0.084	0.070	0.057	0.043	0.028	0.014

table, list of vocabulary, and the tedious research associated with the compilation of all the data necessary.

What we would like to propose instead, is to use the logical research already conducted on the English language: namely its grammatical composition.

As a very simple example, let us construct an input/transformation/output type model for simple sentences. We let the subject be the input: either a noun (possibly with its article) or a subjective pronoun. The transformation would be analogous to the verb, auxiliary verb plus main verb, or a combination of those with adverbs. The output may be a simple noun qualified with as many adjectives as one wishes. The functional relationships may be summarized as shown on the Tableau below (Figure 5.18).

In order to synthesize a sentence, we may pick a representative book in the field we wish to discuss and try to find the vocabulary that is required by the sequence in the Tableau. The fan-in relation is OR, i.e. we only need one word, the rule is to take the first word that corresponds to the requirement, and the fan-out is also an OR type relation where only one word is needed.

We shall list three examples from Shannon's own book (1963) and three from Parkinson's Law (Parkinson, 1964). In the first case, the first word of the page has been chosen, in the second book, the first word after the fifth line in each page. Apart from the

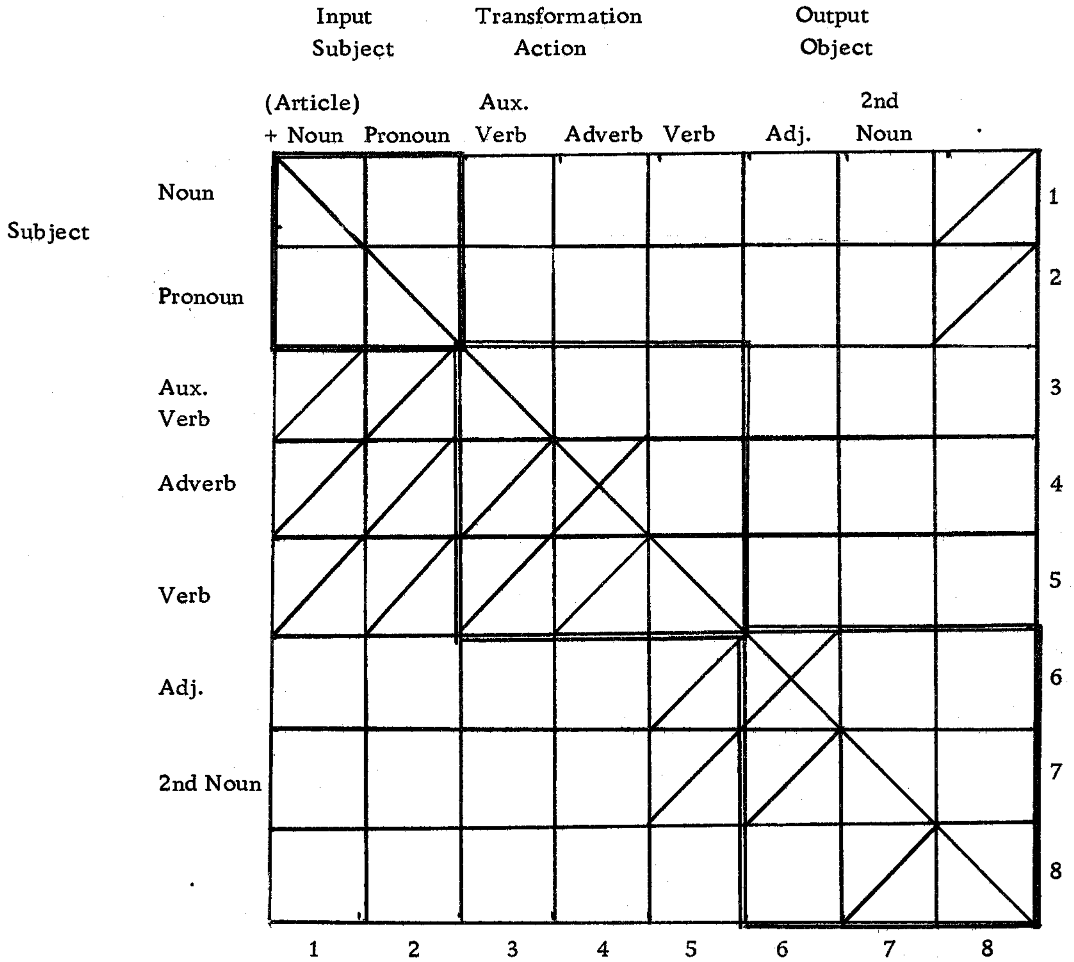


Figure 5. 18. Grammatical Tableau.

obvious grammatical discrepencies (singular with plural, etc.) the sentence does reflect the vocabulary of the field from which the book has been taken (Table 5. 2).

Table 5. 2. Tableau-fabricated sentences.

Shannon's Text

Example 1.

Page	7	8	9	-
Element	1	5	7	8
Word	Teletype	is	dot	.

Example 2.

Page	10	11	12	13	14	15	-
Element	2	5	6	6	6	7	8
Word	We	Suppose	typical	artificial	third-order	letter	.

Example 3.

Page	56	57	58	59	60	-
Element	2	3	5	6	7	8
Word	This	Similarly	change	White	surfaces	.

Parkinson's Examples

Example 4.

Page	15	16	17	18	19	-
Element	1	3	6	6	7	8
Word	leisure	does	reflect	own	recruitment	.

Example 5.

Page	20	21	22	24	-
Element	2	5	6	7	8
Word	he	begun	2000	dockyard	.

Page 23 is a pictorial illustration. Same with page 65 below.

Example 6.

Page	61	62	63	64	66	67
Element	1	2	4	5	6	6
Word	formula	do	constantly	ask	second	candidate's

Page	68	-
Element	7	8
Word	majority	.

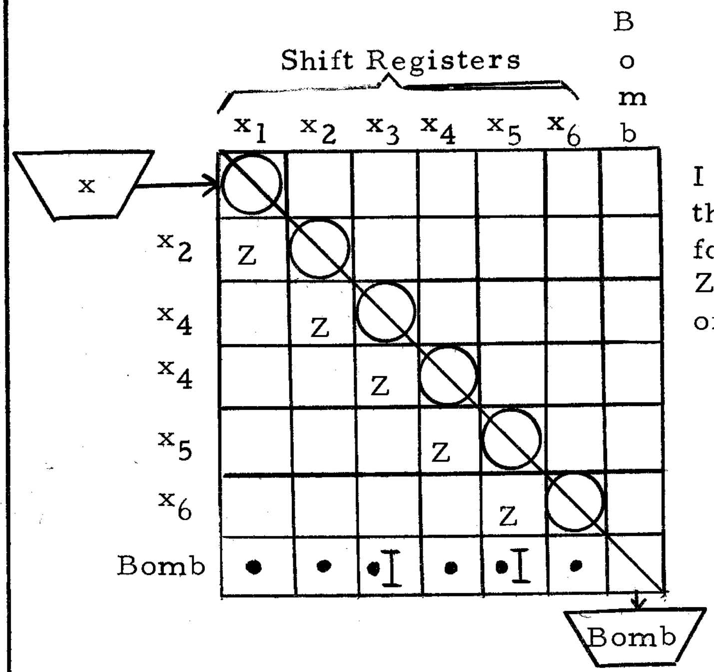
Pattern recognition through State Tableau. In the previous example, we have introduced the concept of a conditional transfer in a Tableau. At each state, there were usually several possibilities of transfer. The transfer, however, was only effected when the given condition has become fulfilled; e.g. an adverb or a verb has been found. The state itself became characterized from the word found at each state.

A traditional approach in describing such a scheme has been called a state diagram. It resembled the usual flowgraph except that the notation (usually a number) on each branch described the condition that had to be fulfilled before the transfer could become effective.

In the preceding sections we have discussed the problem of transmitting a simple code (0 or 1) through a channel. Let us now apply the concept of state diagram to a more complex problem of recognizing a pattern.

Let us suppose for example that the same transmitter we have been discussing was used to send a very special code 101011. It may, for example, mean that an H-bomb should be dropped at a certain target. The transmitter is being used to transmit other messages except in emergencies when this special code is sent. How can we make sure that this particular code is deciphered correctly from the remaining codes of relative unimportant message content?

One possible method would be to save each sequence of six digits and store them in a memory (say a chain of shift registers). Each sequence of six digits can then be decoded in the same way as we have done with the Hamming's code. This is shown in Figure 5. 19 below. (Note that each eigen-cell is made into a register.)



I stands for an inverter that yields 0 for 1 and 1 for 0, or $\bar{x}$ from x.
Z stands for a delayor of a unit time.

Figure 5. 19. Shift register recognition of 101011 pattern.

The other method (Takahashi, 1966), which could be much more logical and efficient would be the use of our state diagram concept. The difficulty associated with the first scheme, that of being caught unaware of the imminant danger may be lessened in the second scheme. In the state tableau approach, the approach of the danger may be detected by the degree of resemblance of the received signal to the critical signal. For example, we may have a buzzer sound off as soon as 10101 has been received to prepare everyone for the possible 101011 signal.

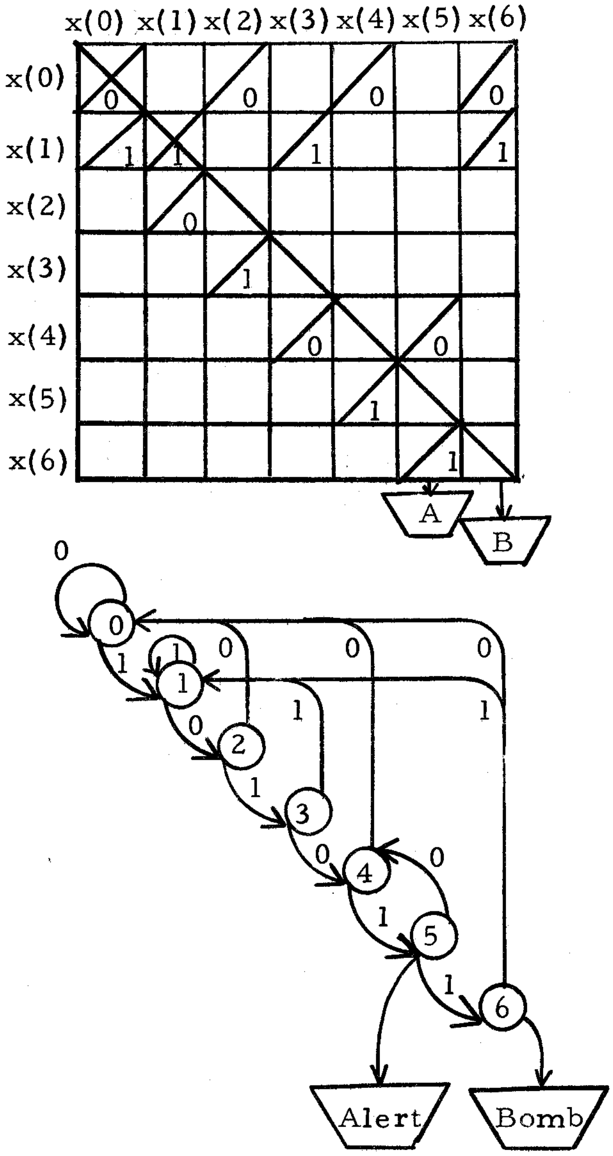


Figure 5. 20. State tableau and state diagram for pattern recognition of 101011.

In the state tableau we shall adopt the convention of writing the condition for transfer (input = 0, S = 0, or simply 0) below the / in each cell (Figure 5.20).

Tableau as a Systems Model

In this chapter, we have examined various means by which a system could be represented by a tableau model.

When the phases of a system were not distinct, an orthogonal set of bases for a new sample state description was thought. The new bases were either a different set of phases, a probability representation, or a parametric representation of statistical distributions.

As examples of "source" models for an information communication model, we have presented the Markov Chain for English language (the Grammatical Tableau), and the Poisson model for random arrivals.

To illustrate the coding and decoding models, we have constructed a Hamming's Code communication model. For the particular case of "Decoder with a memory," we have examined two methods by which the "pattern" of a received signal may be decoded.

The concept of channel as a time-consuming and a continuously time-dependent model was introduced by queueing models. Using an Erlang model, we have also shown how one process may be interpreted in terms of stages.

The limitation of tableau method is precisely that imposed by our ability to formulate an amendable model. Some of the techniques have been illustrated in this chapter. Others are available and we hope more are forthcoming in the future.

> The considerations involved in selecting possible applications are these. First, can the system be adequately described by a number of states small enough to make the solution of the corresponding simultaneous equations computationally feasible? Second, are the data necessary to describe the alternatives of the system available? If the answers to these questions are affirmative, then a possible application has been discovered. There is every reason to believe that a possible application when combined with diligent work will yield a successful application (Howard, 1960, p. 124).

The remaining of this treatise will treat the various methods presently available for solving systems problems. The next chapter will deal with the subject of alternatives in decision making.

VI. DECISION TABLEAU

> Management is the process of converting information into action. The conversion process we call decision making. Decision making is in turn controlled by various explicit and implicit policies of behavior (Forrester, 1962).

Decision-making in a Complex System

When the system under study is complex, a decision can seldom be made at one level. Rather, it usually involves several smaller decisions leading up to the final decision that determines the overall outcome. The levels may be stages in time, such as in sequential decision-making, a hierarchy in an organization, or a combination of both. By the term "sequential," we usually understand that the decision itself is made in sequence rather than the actions resulting from it. For example, a strategic decision of bombing North Vietnam, for example, could be made in a sequential manner, even though the bombing itself may occur only once. If a policy is changed because of the outcome of an experiment, we shall regard this as a reevaluation of the model used, rather than the change in analytic procedure.

In this chapter, we would like to start with an example in which we learn to maximize the information obtainable from a series of experiments; then we shall introduce the concept of an algorithm as a way to establish a "policy," and finally we would like to formulate our decision-making process as a dynamic programming problem that can be generalized for other applications.

Maximizing Information

Discrete Source, Fixed Number of Possible Outcomes

Theorem 6.1. Maximum information. The information generated by a discrete source with a fixed number n of outcomes is maximized when the n outcomes are made equally likely.

Problem statement. The amount of information that can be obtained from such an experiment is expressed by the entropy:

$$H = \sum_{i=1}^{n} h_i = \sum_{i=1}^{n} -p_i \log_2 p_i$$

where p_i must satisfy the condition that:

$$\sum_{i=1}^{n} p_i = 1$$

In order to find the maximum (an extremum) for this entropy H, it seems most natural to employ Lagrange's Multiplier Method.

Lagrange's method of multipliers (Courant and Hilbert, 1953).

> If the variables are not independent but are subject to the restrictions $g_1(x, y, \ldots) = 0$, $g_2(x, y, \ldots) = 0$, $\ldots$, $g_h(x, y, \ldots) = 0$, we obtain necessary conditions for an extremum or stationary point by means of Lagrange's method of multipliers. This method consists in the following procedure: In order to find a point in the interior of the domain of the independent variables at

which $f(x, y, \ldots)$ has an extremum or is merely stationary, we introduce h+1 new parameters, the 'multipliers,' $\lambda_0, \lambda_1, \ldots, \lambda_h$ and construct the function $F = \lambda_0 f + \lambda_1 g_1 + \lambda_2 g_2 + \ldots + \lambda_h g_h$. We now determine the quantities $x_0, y_0, \ldots$ and the ratios of $\lambda_0, \lambda_1, \ldots, \lambda_h$, from equations

$$\frac{\partial F}{\partial x} = 0, \frac{\partial F}{y} = 0, \ldots$$

$$\frac{\partial F}{\partial \lambda_1} = g_1 = 0, \ldots, \frac{\partial F}{\partial \lambda_h} = g_h = 0,$$

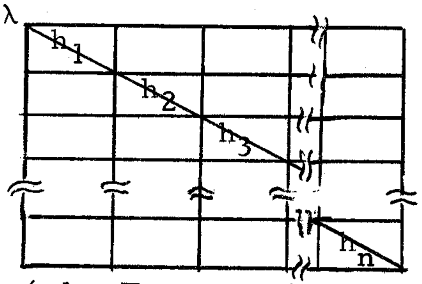
the number of which is equal to the number of unknowns. These equations represent the desired conditions for the stationary character of $f(x, y, \ldots)$ or the extremum of f under the given restrictions. If $\lambda_0 \neq 0$ we may (and shall) put $\lambda_0 = 1$ because F is homogeneous in the quantities λ_i. The Lagrange method is simply a device which preserving the symmetry, avoids explicit elimination of h of the variables from the function $f(x, y, \ldots)$ by means of the subsidiary restrictions (Courant and Hilbert, 1953, pp. 164-165).

Proof of Theorem 6.1. Applying Lagrange's method, we obtain:

$$F = H + \lambda \left(\sum_{i=1}^{n} p_i - 1\right)$$

$$= \sum_{i=1}^{n} (\lambda - \log_2 p_i) p_i - \lambda$$

In passing, we note that this is an eigenvalue problem for the canonized matrix corresponding to the Tableau of Information as shown in Figure 6.1.



$$\frac{\partial (H - \lambda I)}{\partial P_i} = 0$$

Eigen-line represents $(-\lambda I)$.

Figure 6.1. Entropy tableau.

$$\frac{\partial F}{\partial p_i} = \frac{\partial}{\partial p_i}(H - \lambda p_i) = (\lambda - \log_2 p_i) - p_i \frac{\partial \log_2 p_i}{\partial p_i} = 0$$

$$= \lambda - \log_2 p_i - 1 = 0$$

Or, $\quad \log_2 p_i = -1 + \lambda$ for all p_i

and $\quad \dfrac{F}{\lambda} = \sum_{i=1}^{n} p_i - 1 = 0$

Therefore, $\quad p_i = 2^{\lambda - 1}$ and $\sum_{i=1}^{n} p_i = n(2^{\lambda - 1}) = 1$

$$\lambda = \log_2(\frac{1}{n}) + 1$$

$$p_i = 2^{\log_2 \frac{1}{n}} = \frac{1}{n}$$

Thus, checking boundary conditions such as:

$$\lim_{p \to o}(-p \log_2 p) = 0$$

we conclude that:

$$\begin{cases} p_i = \frac{1}{n} \text{ for all } p_i \\ H = \log_2 n \end{cases}$$

Q.E.D.

The resulting extremum can be verified quickly for n = 2 by looking at Table 5.1 or Figure 6.2 below.

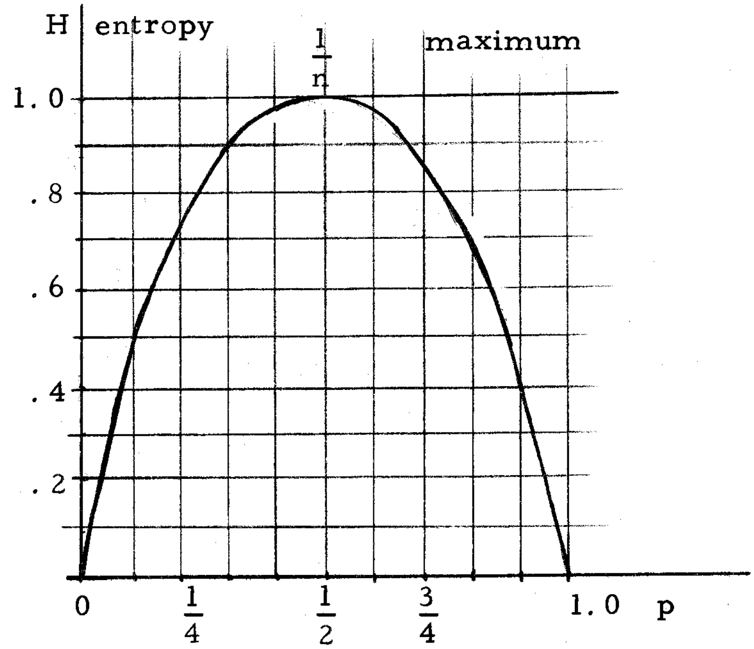


Figure 6.2. Entropy of a binary signal with probability of occurrence = p.

XXR(12-ball Puzzle)

Example Problem 6.1. We are given a balancing scale (without weights) and 12 identical balls with exactly the same weight except for one which may be lighter or heavier than the others. We shall dubb this problem as an "Executive Execution Routine" or an XXR, since it was reported to have cost the career of a computer manufacturer's vice-president who became upset upon being asked to solve this problem (private communication from Dr. J. L. Riggs, 1966).

We shall now examine how the defective ball can be detected in three weighings, together with the information as to whether it is lighter or heavier than the others.

Feasibility study. The information that can be obtained from each weighing of the balance is one of the three following: (1) the pan on the left is heavier than the pan on the right, (2) both pans are weighing the same, or (3) the pan on the right weighs more. We suppose that the balance has been adjusted beforehand and that the pans may contain as many balls as we wish.

The most information we can obtain from one weighing of the balance, according to our Theorem 6. 1, is when the three probabilities corresponding to the above three conditions are equal, or 1/3 each. According to Table 5. 1, this corresponds to:

$$\log_2 3 = 1.585 \text{ bits.}$$

With two weighings, this maximum information will increase to: $2 \times 1.585 = 3.167$ bits $= \log_2 3 \times 3 = \log_2 9$, as you may verify from our Figure 6. 2. With three total weighings, the maximum is now 4. 755 bits.

Having 12 balls of which one is either lighter or heavier than the others (standard = S will be used to indicate normal balls) means that there are 24 possibilities. The total entropy for this is: $\log_2 24 = 4.585$ bits from Table 5. 1.

Since $4.585 < 4.755$, the problem seems solvable.

First tableau. We shall first label the 12 balls as A, B, C, ..., L. We know that we must equalize the probabilities of the three outcomes in order to obtain most information. We shall divide the 12 balls into three groups, n, n, and 12-2n. If we put the first n on one side of the scale and the next n on the other pan, the probability of tilting to either side will be:

$$p_< = p_< = \frac{n}{12}$$

On the other hand, the probability of not tilting to either side is the probability that the defective ball is not in either pan, or

$$p_= = \frac{12-2n}{12} .$$

Equating the three probabilities, we obtain: $\frac{n}{12} = \frac{12-2n}{12}$ or $n = 4$ as the solution. We shall tentatively identify the balls on the left pan as ABCD and those on the right as EFGH. The tableau in Figure 6.3 shows the three possible states. If the balance tilts to its left, we may conclude that *either* A, B, C, or D is heavier than the others, *or* E, F, G, or H is lighter than the others. The total amount of uncertainty is $n = 8$, or 3.000 bits. If the scale balances, then one of the four: I, J, K, or L must be either heavier or lighter than the rest. Thus, $n = 8$ again gives 3.000 bits. The last case is similar to the first with 3.000 bits.

			I	a	b	c
	A, B, C, D, E, F, G, H, I, J, K, L					
	START					
I	(n=24) 4.585 bits 4.755 ABCD vs. EFGH 3 weighings 4.755 bits					
a	2 wighings 3.167 bits > h: A, B, C, D, l: E, F, G, H n=8 3 bits 3.167 bits					
b	= h: I, J, K, L l: I, J, K, L n=8 3 bits 3.167					
c	< h: E, F, G, H l: A, B, C, D n=8 3 bits 3.167					
	INPUT	OUTPUT		h: ABCD l: EFGH	h: IJKL l: IJKL	h: EFGH l: ABCD

Figure 6.3. First-balance tableau for 12-ball puzzle.

Second tableau. It is now possible to construct a tableau for each of the three possible outcomes from the first weighing. However, we shall construct one tableau containing all states for demonstration purposes (see Figure 6.4 and compare this with Figure 6.3 and Figure 6.5).

In essence, we have 3.167 bits available and the uncertainty is only three bits. All we need is to find a combination of three as before, that will give no result that contains more than 1.585 bits of uncertainty. 1.585, as you recall from page 161, is the amount of information that our last weighing can give us at most.

Weighing ABE against CEF, for example in Ia of Figure 6.4, will lead to three possible outcomes with uncertainties of 1.585, 1.000, and 1.585 bits each. This is an acceptable choice, since the largest is equal to 1.585, the amount of uncertainty that can be removed at the last weighing.

Similar reasonings will lead to the tableau in Figure 6.4.

Last tableau. Making the last choices is very easy. Again, it is the matter of balancing the three possibilities to have as close to the uniform probability as possible. The results are shown in Figure 6.5. It is to be noted that we could have built a smaller tableau for each case rather than to have to deal with a huge tableau as the one shown. However, the interrelationships between each stage becomes very clear by using one tableau. For example, the

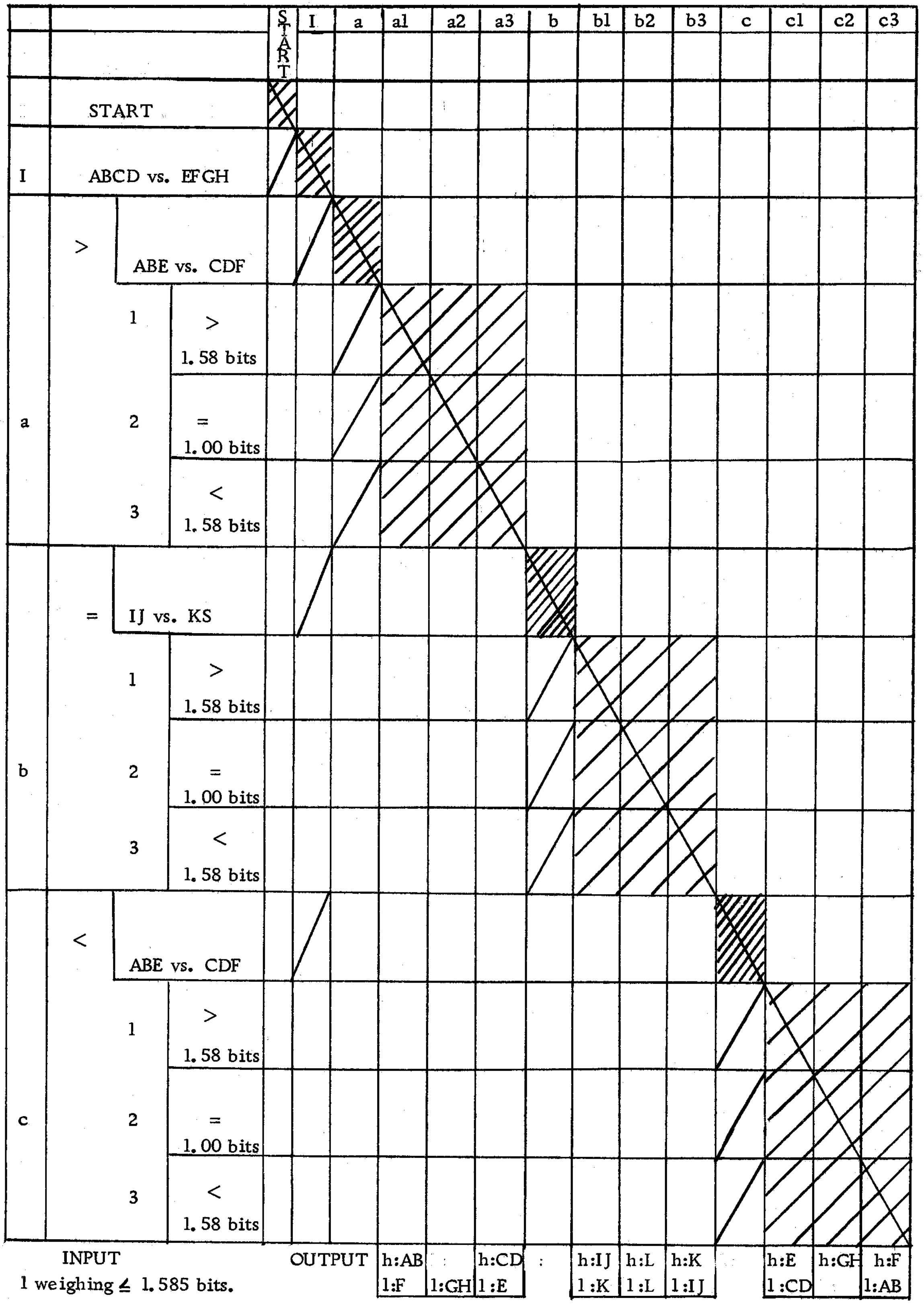


Figure 6.4. Second tableau for XXR.

	Input	Output
	Start	
I	ABCD vs.EFGH	
Ia	>	
Ib	=	?:IJKL
Ic	<	
a	ABE vs. CDF	
a1	>	hAB 1F
a2	=	1:G, H
a3	<	hCD 1:E
1	AF vs. SS	
ii	>	h:A
iii	=	h:B
iiii	<	1:F
2	G vs. H	
2i	>	h:G
2ii	<	1:H
3	CE vs. SS	
3i	>	h:C
3ii	=	h:D
3iii	<	1:E
b	IJ vs. KS	
b1	>	h:IJ 1:K
b2	=	h:L 1:L
b3	<	h:K 1:IJ
1	I vs. J	
1i	>	h:I
1ii	=	1:K
1iii	<	h:J
2	L vs. S	
2i	>	h:L
2ii	<	1:L
3	I vs. J	
3i	>	1:J
3ii	=	h:K
3iii	<	1:I
C	ABE vs. CDF	
C1	>	h:E 1:CD
C2	=	h:G, H
C3	<	h:F 1:AB
1	CE vs. SS	
1i	>	h:E
1ii	=	1:D
1iii	<	1:C
2	G vs. H	
2i	>	h:G
2ii	<	h:H
3	AF vs. SS	
3i	>	h:F
3ii	=	1:B
3iii	<	1:A

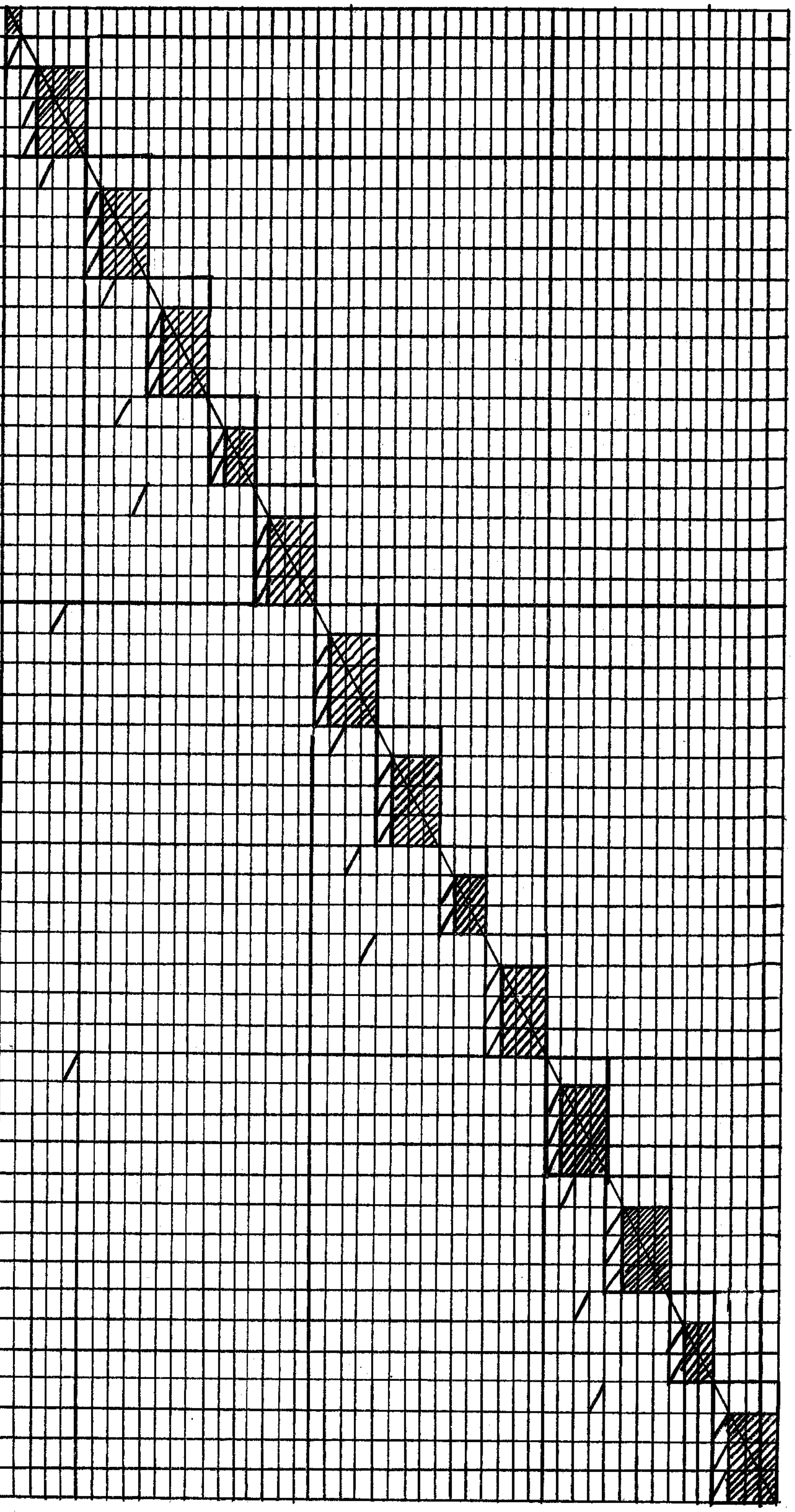


Figure 6.5. Third tableau for XXR.

state a in Figure 6.3 became a tableau with 16 cells in Figure 6.4, each of which was again expanded into 9 to 16 cells in Figure 6.5 (see Figure 6.6).

Though this may not be a very practical example (unless you happen to be a vice-president of a computer manufacturer) it does point out the typical features of a Discrete Source Fixed Number of Possible Outcomes Experiments.

We shall next tackle another "game" in which the opponent is supposed to be using his wit as fully as we are. Then, in the next chapter we shall treat the problems where stochastic forces (noises) are disturbing our systems.

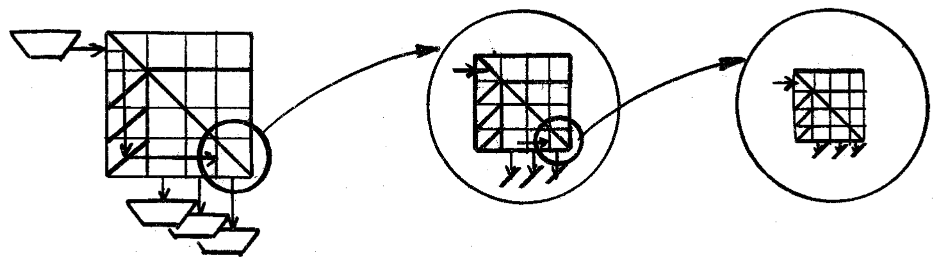


Figure 6.6. The expansion of tableaux.

Algorithmic Approach

Algorithm

Etymology. The simplest and best known algorithms are the four arithmetic operations using Arabic numbers. The term

"Algorithm" is usually attributed to the Arabic (Uzbek) mathematician, Al-Khowarāzmī, whose name literally means "native of Khwārazan (or Khiva)" and who is said to have given such rules in decimal system as early as the ninth century.

Definition 6. 2. Algorithm. An algorithm is a list of instructions specifying a sequence of operations which will give the answer to any problem of a given type (Trakhtenbrot, 1963: original Russian edition appeared in 1960).

Using another simple game (6. 2) of the type we have just discussed (6. 1), but using match sticks instead of balls, we shall try to see how an algorithm can be found, and how they can be related to the concept of "policy."

Six-match Problem

Example Problem 6. 2. Two players are sitting at a table upon which six objects (say matches) have been placed. Each is allowed to pick either one or two pieces at a time, playing alternately. The one who is to pick the last piece is the loser.

Our objective is to find a scheme, or an algorithm, for winning the game (that is, if possible).

If this game seems too trivial, it is for the sake of illustrating concepts with a minimum of useless complications. However, for those interested, we shall show a 24-match game with three possible moves (take 1, 2, or 3 matches each time) and a way to expand it to any size game, at the end of this section.

Properties of this game. Let us now examine some of the peculiar features of this game:

1. Two-person game: this game involves two players each playing alternately until one loses.

2. Dichotomic ending: either the first or the second player wins at the end of a game. There is no draw.

3. Complete knowledge: each player is given complete information on all past moves and what he is allowed to do next.

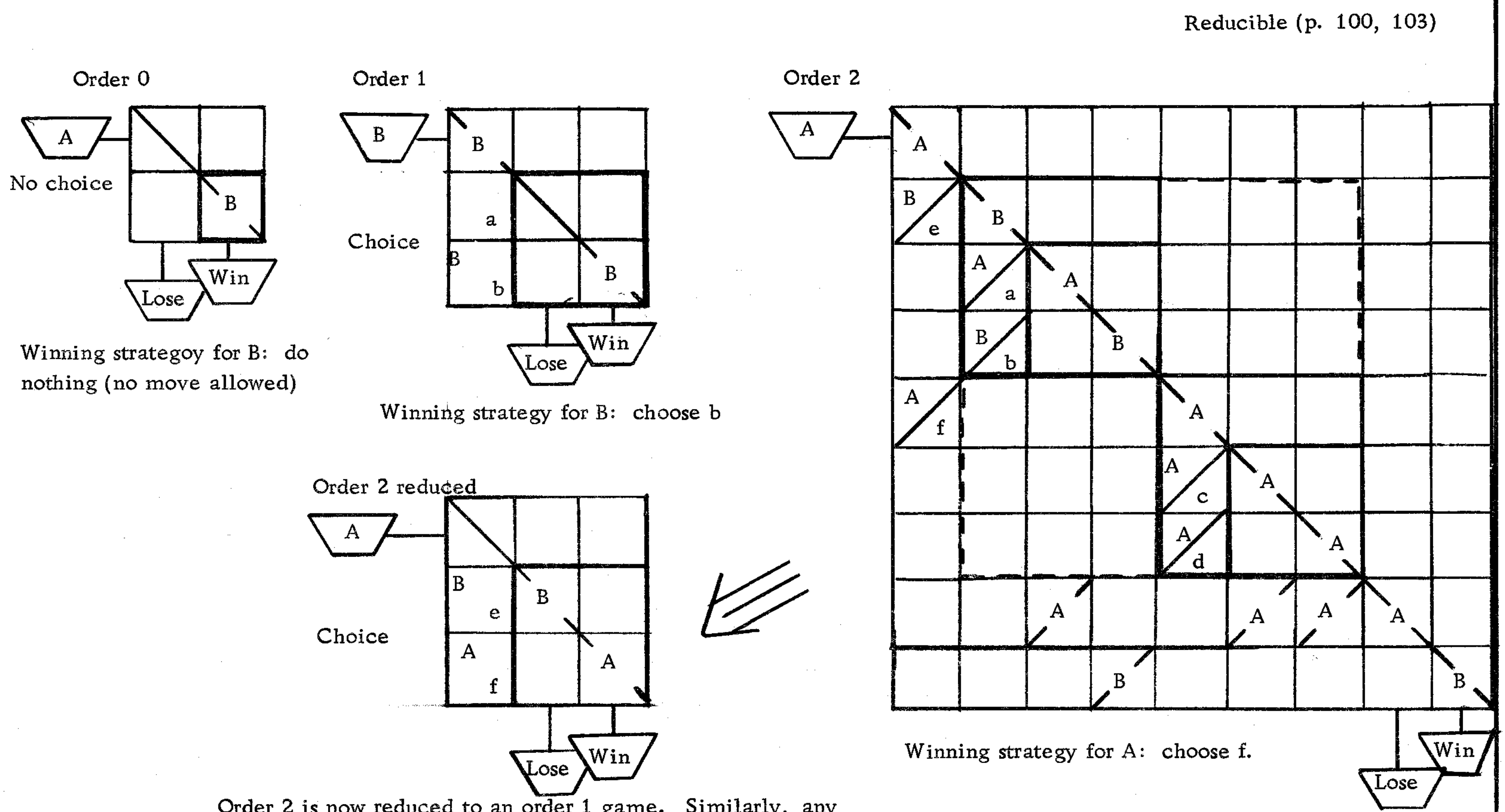
4. Free choice: each player can select any move (u) from the set (U) of moves composed of all allowed plays: $u \in U$.

5. Terminal game: the game always terminates within a given number of moves. This upper limit of the number of moves (levels) is called the order of the game. The order of the six-match game is obviously six, corresponding to the slowest game in which each player will take only one match at a time.

Theorem 6.2. Existence of a winning strategy. In any game satisfying properties 1-5 listed above, there is a winning strategy for one of the players.

Proof of Theorem 6.2. In order to prove this theorem, we must use induction to show that an algorithm can be constructed for a game of any order. Let us start with a game of order zero. In this case, the game is predetermined: either the first player (A) is a winner, or the second player (B) is a winner, and there is nothing (no move) they can do about it (strategy for the winner: do nothing). In the order is one, one player is allowed to make one move. After he has made one move, the game is reduced to that of order zero, and a winner is decided. Thus, there is a definite strategy when the order is one. Of course, the winner is not necessarily the one making the move. To go from a game of order n to n + 1, we simply note that having a winning strategy is as good as winning a game for a player intelligent enough to follow the strategy faithfully. If the player fails to win the game after following his winning strategy, his strategy cannot be called a winning strategy. Therefore, by considering the game at stage n when there exists a winning strategy to be equivalent of a game of order zero, where a winner exists for certain, a game at n + 1 is reduced to that of order 1 when one move will lead to order zero. Continuing this process, we can expand the game from order n to n + 1 to n + 2, *ad infinitum*. This concept is best shown using a tableau illustration as in Figure 6.7.

Algorithm for the match-game. Figure 6.8 is a tableau corresponding to the six-match game. For each state, marked by the player making the move (A or B) and the number of matches on the table (6, 5, 4, 3, 2, or 1), there are two alternatives (decision 1 to



Order 2 is now reduced to an order 1 game. Similarly, any order n + 1 can be reduced to order 1 for which there is a winning strategy (provided n had a winning strategy).

Figure 6. 7. Reduction of order for winning strategies.

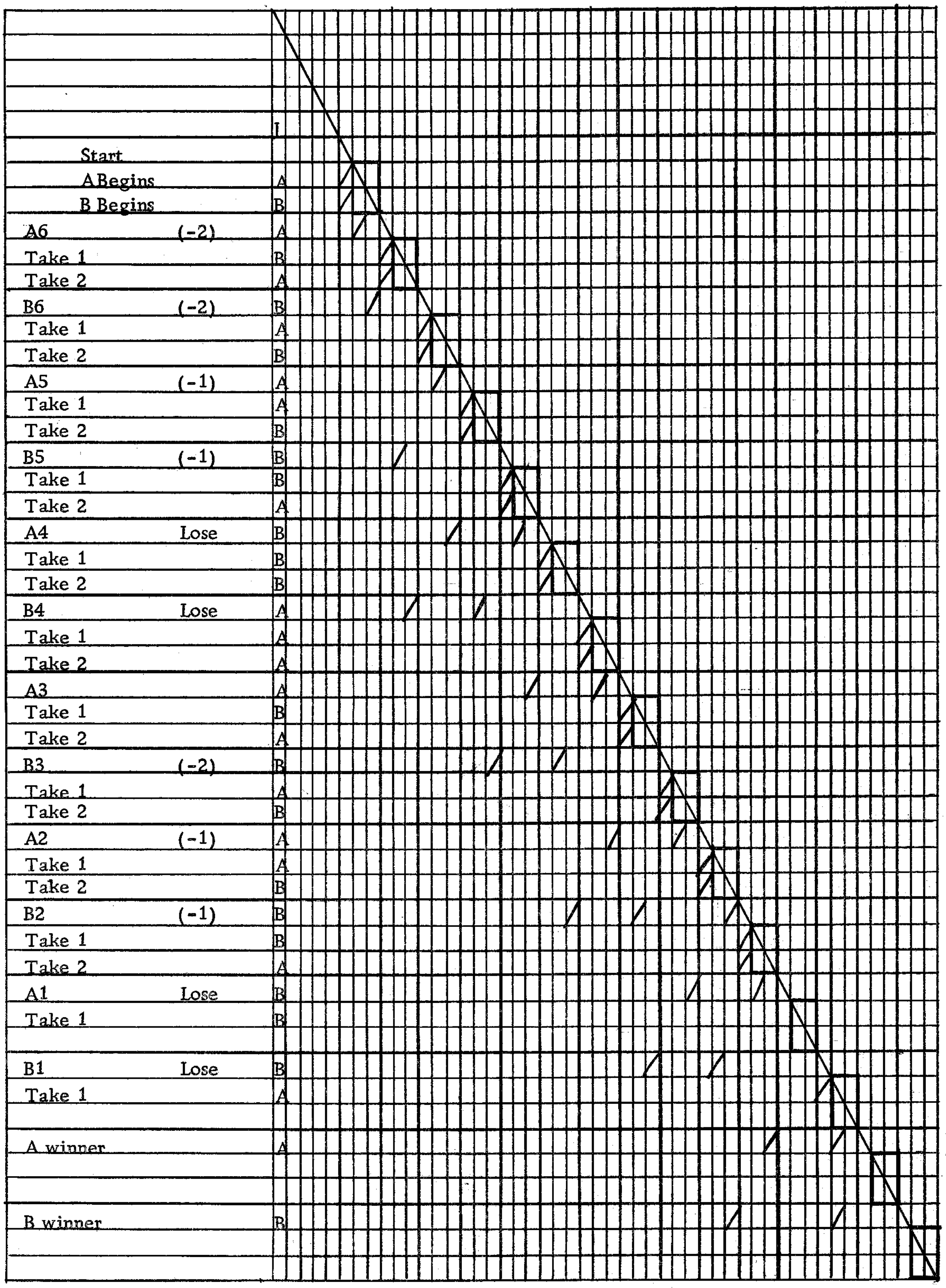


Figure 6.8. Six-match game tableau.

take 1 match, or decision 2 to take 2 matches) except for the last states where there is only one possible move (take 1 and lose the game).

We assume that each player wants to win, and will choose the winning strategy whenever one is available. Then working back from the last level, corresponding to a game or order 0, we can work backward toward the higher levels by marking the winner (A or B) for each strategy by placing A or B inside the eigen-cell. Similarly, to facilitate the observation, we have marked each transducer with the winner in the next stage (state). In addition, the expression (-1) or (-2) indicates the winning strategy at each level (state).

From the observation of the tableau, we note that at A4, B4, A1, and B1, the player making the move has no winning strategy. The trick of the game is to force the opponent into one of these states and to keep him in those "lose" states until the end of the game. Thus, if A is starting at A6, he will take (-2) alternative of removing two matches forcing B into stage B4. Whatever B does, say by taking u numbers of matches ($u \in U \{1, 2\}$), A can force B back into another "lose" state by removing (3-u) number of matches (B1). This is the essence of the algorithm for winning the game. The explicit formulation of this algorithm is what we usually call a "policy." Obviosly, the first item in the policy is not to start the game unless you are the first player.

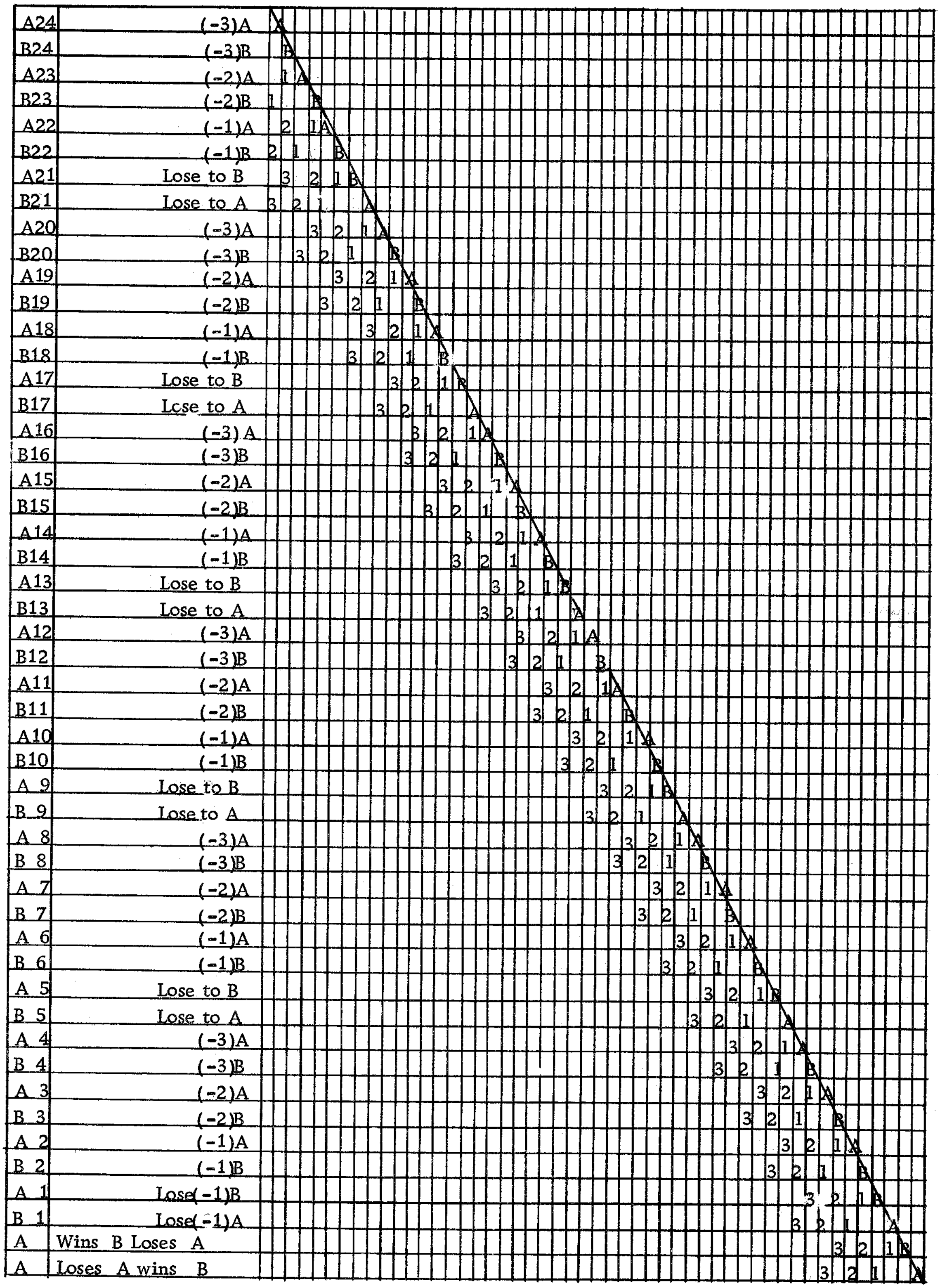
<u>24-match game</u>. The analogy can be extended to winning a 24-match game with three alternatives ($u \in U$: $\{1, 2, 3\}$), or to any number of matches with as large alternative set as one wishes. Figure 6.9 is a tableau for the 24-match case. Note that the tableau has been reduced to show only the states and not alternatives (except as transducers).

If there are m number of matches, and each player is allowed to take up to and including $\bar{u}$ number of matches at a time, the "lose" states will occur anytime when there are $n\bar{u} + 1$ number of matches left ($n = 0, 1, 2, \ldots < \frac{m}{u}$). For example, this means 1, 4 for the 6 matches, and 1, 5, 9, 13, 17, 21 for 24 matches.

The winning strategy, thus its algorithm, is to first take ($m - \tilde{n}\bar{u} - 1$) matches where $\tilde{n}$ is the highest possible n. Then, from the next move on, always take $\bar{u} + 1 - x$ number of matches, where x is the number taken by the opponent at his previous move.

If you wish to make the game such that the player starting the game will lose the game, it is only necessary to make $m = n\bar{u} + 1$.

<u>Generalizations</u>. We can further extend our Theorem 6.2 to cover cases where properties 1 and 2 do not hold but where 3, 4, and 5 are still binding. In other words, we may consider cases where there are more than two players, or where a tie is allowed. By placing a suitable limit (say 20 moves) and allowing "tie" as a possible outcome, it is theoretically possible to build a tableau that



*All figures in the tableau should be read as negative terms.

Figure 6.9. 24-match game tableau.

will yield a winning strategy (or at least a tying one) for one of the chess players. A similar comment may be made for the game of checkers. There are only a finite number of games that can be played in either case.

> In checkers, the total number of continuations [(a probable misprint of 'configurations')] is in the order of 10^{40}. Dr. A. L. Samuel points out that even if three choices could be examined in every one-thousandth of a second, it would take 10^{21} centuries for a computer to consider all of these possibilities. Shannon estimated that there are in the magnitude of about 10^{120} chess games which can be played (Desmonde, 1964).

Once an algorithm is found, we consider that the particular set of problems have been solved. For example, the fact that we have never added 1.41421356, 1.7320504, and 2.2360679 together does not make us hesitate in considering the problem as good as being solved, that is if we were mathematicians. Of course, if you are an engineer wishing to use this particular number, knowing the algorithm may only be the beginning of a long tedious task of finding the answer. (In Chapters I and III we have discussed the marvel of our number system which allows us to operate on as large or as small numbers as we wish.)

Aside from the computational hardships, there are more fundamental difficulties. Mathematicians have been struggling for ages to find an algorithm that will find other algorithms, or at least prove the existence thereof. For example, David Hilbert proposed the famous "Hilbert's Tenth Problem" based on his axiomatic approach: "Find an algorithm for determining whether any given Diophantine equation has an integral solution" (Trakhtenbrot, 1960, p. 6). A Diophantine equation is a polynomial of the form: $P = 0$, such as: $x + 2x^2 + 4x^3 + 8x^4 = 0$, or $x^2 + y^2 + z^2 = 0$ where integral solutions were sought for polynomials with integral coefficients.

If we consider x as a mapping operation, the first of these reminds us of a chained transducer in our tableau. The temptation becomes very great to see whether we could possibly formulate a tableau that will tell us whether a tableau can solve a particular problem. This is where we must remind ourselves of Gödel's Incompleteness Dilemma (Chapters I and III). If we could have built such a tableau, either our tableau method is inconsistent or our result unreliable.

Then, is it possible to arrive at an algorithm with a tableau as long as we stay within the scope of a tableau? Again, unfortunately the answer is no. We have based our method on the theory of Group and used the identity element as the basis for our eigen-line.

> In 1955 P. S. Novikov created a great stir in the mathematical world by demonstrating the algorithmic unsolvability of the identity problem in group theory ... The existence of the admissible substitutions a $\alpha \rightarrow e$ then means that for every elementary transformation 'a,' there exists an elementary transformation α such that the application of 'a' followed by α is the identity transformation ... Novikov constructed an example of a calculus satisfying the group

> axiom, for which no algorithm exists. Therefore, a general axiom for all groups is likewise impossible (Trakhtenbrot, 1960, pp. 89-90).

Suddenly, it appears as though our tableau method is not nearly as powerful of a tool as we have originally wished. However, if the limitation exists, it is one that no other known tool has overcome.

Before proceeding with the question of decision, we present another little game; this time it is a riddle from W. Ross Ashby, which we shall dubb "Ashby's ghosts."

Ashby's Ghosts

Example Problem 6.3.

"Graveside"
Wit's End
Haunts.

Dear Friend,

Some time ago I bought this old house, but found it to be haunted by two ghostly noises--a ribald Singing and a sardonic Laughter. As a result it is hardly habitable. There is hope, however, for by actual testing I have found that their behaviour is subject to certain laws, obscure but infallible, and that they can be affected by my playing the organ or burning incense.

In each minute, each noise is either sounding or silent--they show no degrees. What each will do during the ensuing minute depends, in the following exact way, on what has been happening during the preceding minute:

The Singing, in the succeeding minute, will go on as it was during the preceding minute (sounding or

> silent) unless there was organ-playing with no Laughter, in which case it will change to the opposite (sounding to silent, or vice versa).
>
> As for the Laughter, if there was incense burning, then it will sound or not according as the Singing was sounding or not (so that the Laughter copies the Singing a minute later). If however there was no incense burning, the Laughter will do the opposite of what the Singing did.
>
> At this minute of writing, the Laughter and Singing are both sounding. Please tell me what manipulations of incense and organ I should make to get the house quiet, and to keep it so (Ashby, 1963, p. 60).

Tableau solution. In order to make Ashby's problem more intriguing, we shall impose the condition that the solution cannot involve the change of two inputs at the same time. After all, it is very difficult to light incense while playing an organ so that it becomes lit the moment you stop your organ music. Also, from the description of the present state (both sounds on), it is obvious that our friend has his incense lit, but no organ, unless he can write a letter while playing his organ (which is another theoretical possibility).

Figure 6.10 shows the tableau solution to Ashby's problem. We identified four states by the vector (x_1, x_2) where x_1 indicated the singing (1 for on, 0 for off), and x_2 indicated the laughter (1 for on, 0 for off). Similarly, the control vector is also composed of two components, u_1 for the organ (1 playing, 0 silent) and u_2 for the

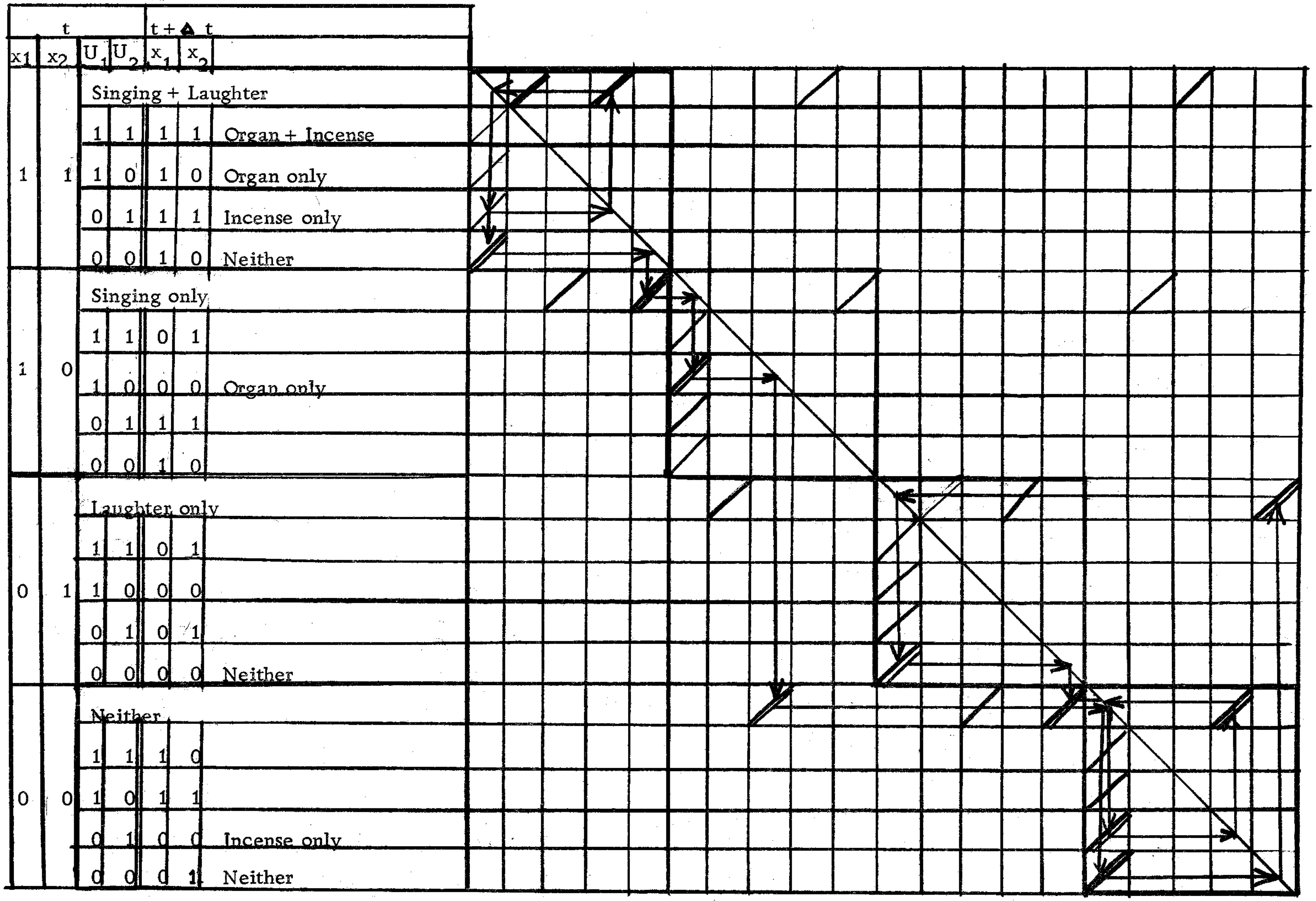


Figure 6.10. Ashby's Ghosts Tableau.

incense (1 burning, 0 extinct). In each state $\underline{X} + (x_1, x_2)$, we have four possible choices for $\underline{u} = (u_1, u_2)$:(0, 0), (0, 1), (1, 0), and (1, 1).[12] With each choice of input, there is only one outcome $\underline{Y}$ at time Δt later, where Δt is a suitable duration of time corresponding to Ashby's "minute."

The solution, as it is obvious from the tableau, is the sequence of inputs, or a policy, that will:

min. 1: extinguish the incense $\underline{u} = (0, 0)$; $\underline{x} = (1, 1)$; $\underline{y} = (1, 0)$

min. 2: play the organ $\underline{u} = (1, 0)$; $\underline{x} = (1, 0)$; $\underline{y} = (0, 0)$

min. 3: stop the organ $\underline{u} = (0, 0)$; $\underline{x} = (0, 0)$; $\underline{y} = (0, 1)$

min. 4: do nothing $\underline{u} = (0, 0)$; $\underline{x} = (0, 1)$; $\underline{y} = (0, 0)$

min. 5: burn the incense $\underline{u} = (0, 1)$; $\underline{x} = (0, 0)$; $\underline{y} = (0, 0$_

min. 6... on: same as in min. 5.

Notice that the state (1, 1) was a self-perpetuating state while the incense was burning. This is the condition that prevailed when the letter was written, and we recognize this particular pattern as the periodic oscillatory form of tableau (p. 120). Similarly, we note that the self-perpetuating condition is used in state (0, 0) to keep both the singing and the laughter from occurring again (min. 5 on). This is another case of a periodic subgroup which i s turned into a sink by prohibiting the system to leave the state. By contrast, we may

[12] Actually only three because of our added restriction (p. 179) of changing only one input at a time.

say that the state (1, 1) was made into a source state (a transient state). Figure 6. 11 shows the reduced forms.

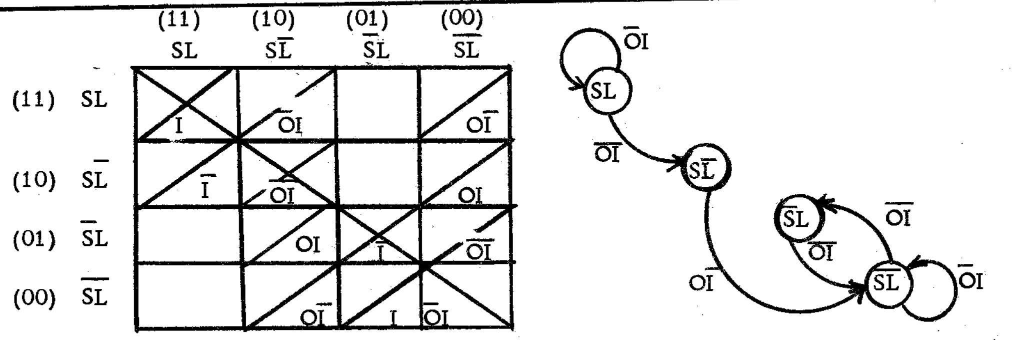


Figure 6. 11. Reduced tableau and flowgraph.

Automaton

Automaton Tableau. By this time, it is becoming rather obvious that whenever we are trying to solve a problem, we are in fact turning tableau into a sort of Turing machine. The fact that a tableau is essentially a finite automaton can be seen clearly if we compare our last example to the Definition 1. 8 on page .

I, the set of input includes (0, 0), (0, 1), (1, 0), and (1, 1) and corresponds to the set of control input U from which one control vector $\underline{u} \in U$ is chosen.

0, the set of output includes (0, 0), (0, 1), (1, 0), and (1, 1) corresponds to our set of $\underline{y} \in X$.

S, the set of internal state is the same set X as that of our output.

Therefore, the next-state function, λ, and the next output function, ∂, are also identical, the output of one state being its condition at the next time interval.

Turing machine. How close is a tableau to a Turing machine? An obvious limitation of a tableau is its dimensionality. In order to have a tableau with a memory comparable to a Turing machine with an infinite tape, the tableau must, in essence, become a Hilbert space (i.e. an orthonormalized space of infinite-dimension, see Courant and Hilbert, 1953, p. 55). However, a Turing machine with an infinite tape is as abstract, in a sense, as a tableau with Hilbert Space representation. Let us take a few moments to examine a Turing machine in more detail. There are several reasons why we would like to discuss now the concept of a Turing machine. We shall give two main reasons.

First of all, the concept of Turing machine had a great impact on Wiener's Cybernetics (Wiener, 1961, pp. 13, 23, 125, 126, etc.) and on the development of modern computers. Turing machine was first described by the English mathematician A. M. Turing in his "On computable numbers, with an application to the Entscheidungsproblem" in the Proceedings of the London Mathematical Society (Series 2, vol. 42), (1936-37, pp. 230-265).

> Turing, who is perhaps the first among those who have studied the logical possibilities of the machine as an intellectual experiment, served the British government

> during the war as a worker in electronics, and is now in charge of the program which the National Physical Laboratory at Teddington has undertaken for the development of computing machines of modern type (Wiener, 1961).

Wiener has visited Teddington to discuss his ideas of Cybernetics with Turing in 1947 prior to the publication of his book on Cybernetics in 1948.

Second reason is the basic hypothesis of the theory of algorithms fostered by Trakhtenbrot (1963, p. 77): "All algorithms can be given in the form of functional matrices and executed by the corresponding Turing machines."

This claim is well substantiated by Trakhtenbrot, and will enable us to generalize our stipulation for the use of tableau in algorithmic problems if we can show sufficiently close similarity between the functioning of the two.

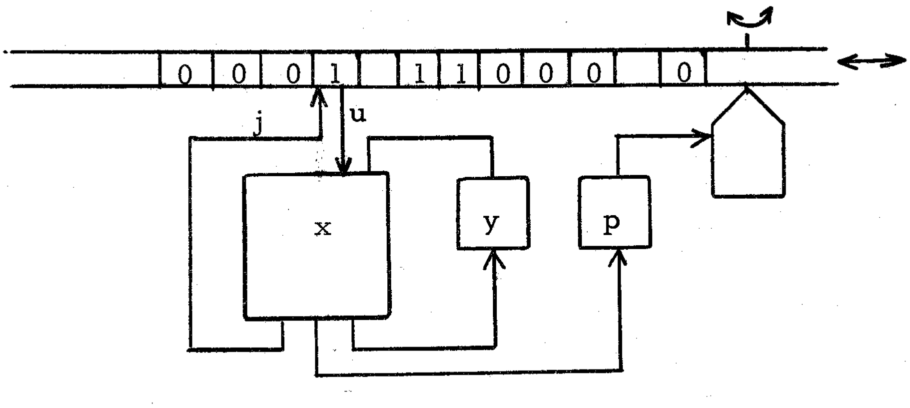


Figure 6.12. A Turing machine.

<u>Turing mechanism.</u> Though a Turing machine is a theoretical

machine which has not been constructed for any practical purpose (some computers have programs simulating a Turing machine), it may be considered as a most rudimentary digital computer coupled to a magnetic tape unit with an infinite length tape.

1. Tape unit. The tape is divided into cells containing one character each (similar to our concept of a byte or a BCD (binary coded decimal character). There is a finite number of symbols (alphabets) that can be used for these letters. The tape can be moved either to the right (R), to the left (L), or kept at the same cell (S). If there is a motion, it can be moved by only one cell at a time (P).

The letter stored in the tape cell being scanned can be read in as an input (u), and depending on its content and the internal state of the machine (x), a new (or the same) character may be returned to the tape cell. The old content of the cell is erased and the new information is entered (J).

2. Machine. The machine itself may be considered as a register whose new state (y) depends on its old state (x) and the new input (u).

3. Functional matrix. The new state, the new symbol, and the new position are determined by the present state and the present input in accordance with a functional matrix provided with each machine. An example of a very simple matrix is shown in Figure

6.13. We have limited ourselves to the choice of three symbols: 0, 1, and blank; and three states: x_o, x_1, and x_2

State x	Input u	Next State y	New Symbol J	Tape Motion* P
x_o	0	x_o	0	R
x_o	1	x_2	0	R
x_o	blank	x_o	blank	S
x_1	0	x_o	0	R
x_1	1	x_1	1	L
x_1	blank	x_1	blank	L
x_2	0	x_1	1	S
x_2	1	x_2	1	R
x_2	blank	x_2	blank	R

* R: Right Cell
L: Left Cell
S: Same Cell

Figure 6.13. An example of a functional matrix for a Turing machine.

Figure 6.14 shows an example of what this matrix will do to a tape containing information as shown in Figure 6.12. For our convenience, we shall illustrate the step-by-step changes by moving the Turing machine rather than the tape. We notice that this particular machine moves binary 1's from left to right so that the number of 1's on the right will be the sum of the number of 1's on the right and left of the blank at the beginning (1 added to 110 gave 111).

Tableau vs. Turing machine. The similarity between a tableau and a Turing machine is rather evident. Figure 6.15 shows

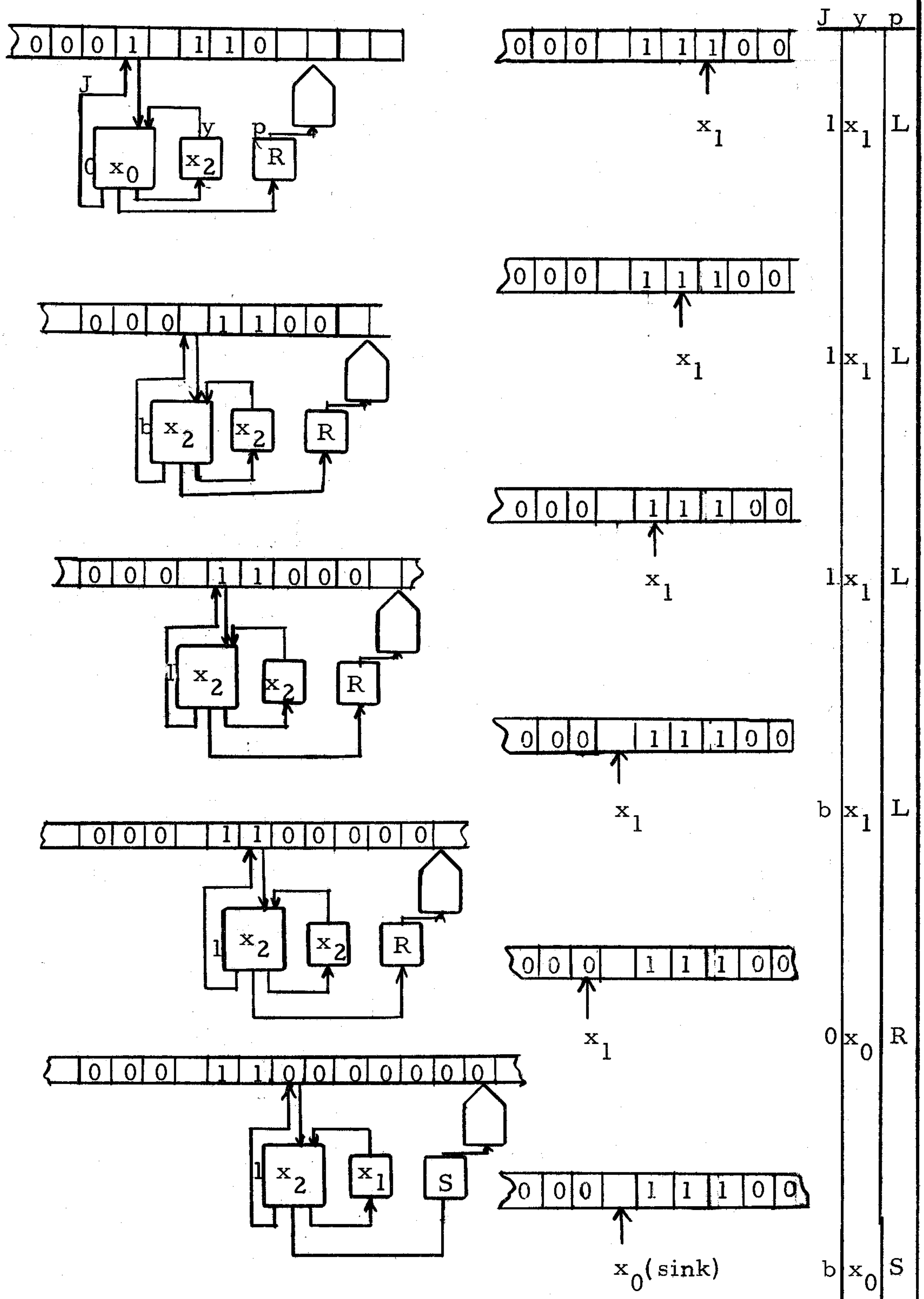


Figure 6.14. A sample operation for a Turing machine.

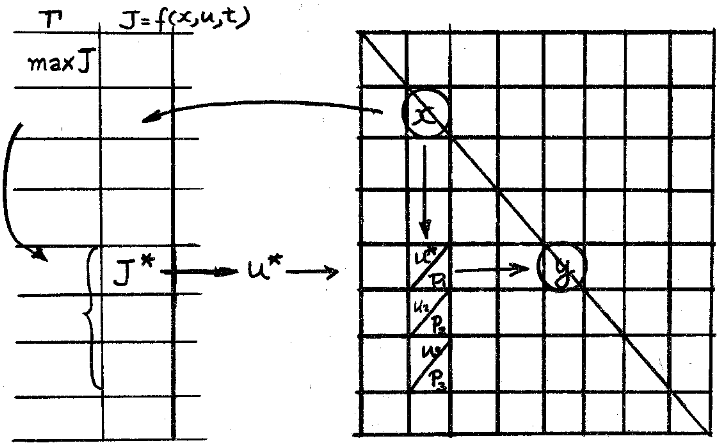


Figure 6.15. A tableau interpretation of a Turing machine.

the tableau's interpretation of a Turing machine. In a tableau, the content of the eigen-cell (which may be listed on the side of the tableau, rather than inside the crowded cell) corresponds to the information on the tape. Depending on the information received(u) and the state(x) of the system, a transducer is chosen that will conditionally (/P) determine the new state(y). How we can use the tableau to update the information will become clear when we shall tackle the Toy-maker's problem in the next chapter. Let us close this section with some quotes:

> ...any process which can be carried out by a digital computer can also be carried out by a Turing machine. ...we are tempted to make the following hypothesis, following Turing in his original 1936 paper: Turing's hypothesis: The informal intuitive notion of an effective procedure on sequences of symbols is identical with our precise concept of one which may be executed by a Turing machine (Arbib, 1964).

Definition 6.3. Recursive function. A function is called recursive if there is an effective procedure for computing it.

> Besides the Turing functional matrices, several other methods were proposed. For example, A. A. Markov arrived at the normal algorithm. . . and Gödel and Kleene arrived at the concept of recursive algorithm (recursive function). It turned out that all these are equivalent (Trakhtenbrot, 1960).

Decision-making

Game Matrix

Comparison of three examples. In this chapter, we have thus far examined three example problems. The first one involving 12 balls was a problem where the outcomes were completely at the mercy of Nature. In the second example of six-match game, we have assumed an intelligent opponent who will always choose a winning strategy whenever one was available. In the last example of Ashby's Ghosts, we (management) had the complete control, and the Ghosts (Nature) were supposed to follow a deterministic pattern without any choice. Obviously, the first and the last are particular cases of the second, and the second problem can be turned into either extreme by specifying appropriate constraints. As a matter of fact, by assuming that our opponent will always choose a winning strategy, we have done precisely that. The problem has become deterministic, since in most cases our opponent would have had only one choice (the winning strategy), and where he had a choice, it did not really matter which one he has chosen (the "lose" cases).

Once the problem has been reduced to a deterministic game, it is usually possible to turn our tableau into a Turing machine to solve the algorithm. The key question is not how to run a Turing machine or use a tableau, which we know already, but how can we construct a deterministic rule or a "functional matrix" that will lead us to the solution. For this, we would like to go back to our six-match problem and examine our strategies more carefully.

Toy Tableau. The assumption we have made in solving the six-match problem can perhaps be best illustrated by considering a toy "computer" that will tell any kid what the winning strategy is for such a game. Figure 6.16 shows a reduced (five-match) version of this "toy." It consists of a grid of wire, nine vertical and nine

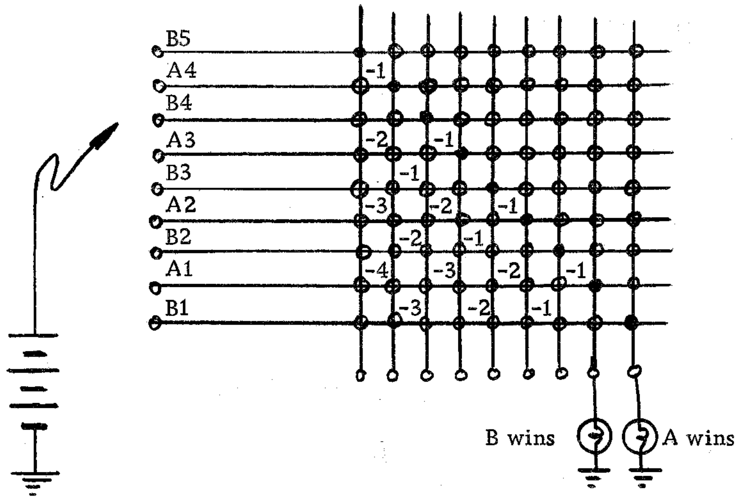


Figure 6.16. Toy Tableau.

horizontal, each vertical wire coupled to one horizontal wire at its eigen cell. The horizontal lines are marked B5, A4, B3, etc. The vertical wires corresponding to A1 and B1 are connected to two light bulbs, one marked "A wins" (B1), and the other "B wins" (A1). The other side of the light bulb is routed through a ground to one side (say negative) of a battery. The positive side of the battery may be connected to any horizontal wire. Obviously, if A1 is touched with this probe "B wins" will light up, while if B1 is touched "A wins" will light up. At the intersections of vertical and horizontal wires, there are female sockets that can be shorted by inserting a male plug. Of course, in order to avoid the reverse flow of signal under illegal conditions, a more expensive version of this toy may have plugs that will connect the horizontal and vertical wires through diodes.

This toy can now be used in the very same manner our tableau was used to find the winning strategy. Instead of placing a / in the appropriate cells corresponding to either the strategy of taking one match or to the one of taking two matches, the male plug can be inserted. If the light bulb that indicates your winning is lit, the strategy is optimal. On the other hand, if the opponent's light lights, obviously it is the wrong strategy. If both light bulbs are on, the strategy cannot guarantee your winning.

The deterministic nature of the game is forced on, by not allowing the conditions in which both lights are on. The opponent is supposed to use only one plug per state and this is to be his winning strategy if one exists. If this law is obeyed, (when you choose a strategy) only one light will light and both lights will never light simultaneously.

Of course, we must assume that a proper procedure has been followed in constructing the winning policy. The strategy must be chosen one at a time starting at B2, then A2, B3, and up. At each stage, the battery probe must be plugged into the appropriate horizontal wire (B2, A2, B3, etc...).

This little toy was originally a part of an analog demonstrator that was used to simulate a Critical Path Scheduling Tableau and was equipped with a potentiometer bridge that measured voltages corresponding to the Earliest Start, Earliest Finish, Latest Start, and Latest Finish Dates for each activity and for the entire project.

The device has successfully demonstrated the theoretical feasibility of such an analog computation but also its practical difficulties (nonlinear voltage drop, Zener effects, etc.).[13]

Two-person game. The deterministic behavior we have assumed, is obviously very unrealistic. Even a genius makes a mistake, and the objective for playing a game is not always so clearly

[13] Turing had his "machine," Babbage had his "engine," and Ashby his "homeostat" (Ashby, 1963, p. 83), none of them workable at the time of their conception. Perhaps we may take our consolation from the fact that we can at least guarantee our toy's workability when it is reduced to the simplicity of a five-match model.

defined either. What do we really mean by "a winning strategy"?

To limit our discussion to the realm of materialism, let us assume that the winner is paid ten dollars from the loser at the end of the game. Also let us identify ourselves with player B so that A is our opponent.

As usual, we start our investigation from the last state. When we are confronted with only one match which must be picked up, we have no choice but to lose the game. We shall say that our game is costing us ten dollars, or $J = -10$. If A1, rather than B1 occurs, we know that our opponent has no choice but to lose the game, and we can confidently say that $J = +10$. So far, our reasoning is "deterministic."[14] When the situation is B2, however, we have two choices, take one and win the game, or take two and lose the game. Obviously, the optimum strategy is to take one for $J^* = +10$, and the winning strategy is the optimum strategy u^* corresponding to (-1). Figure 6.17 illustrates our one-sided (player B's) viewpoint.

At stage B3, the picture is no longer as clear-cut as before. For certain, we do have a winning strategy $u^* = (-2)$, but the outcome of what happens when we choose $u = (-1)$ depends on A's next move. If he takes $u = (-1)$, we lose with $J = -10$. If he takes $u = (-2)$, on the

[14] This corresponds to the case where "A win" bulb lights up in our Toy Tableau regardless of the alternative chosen.

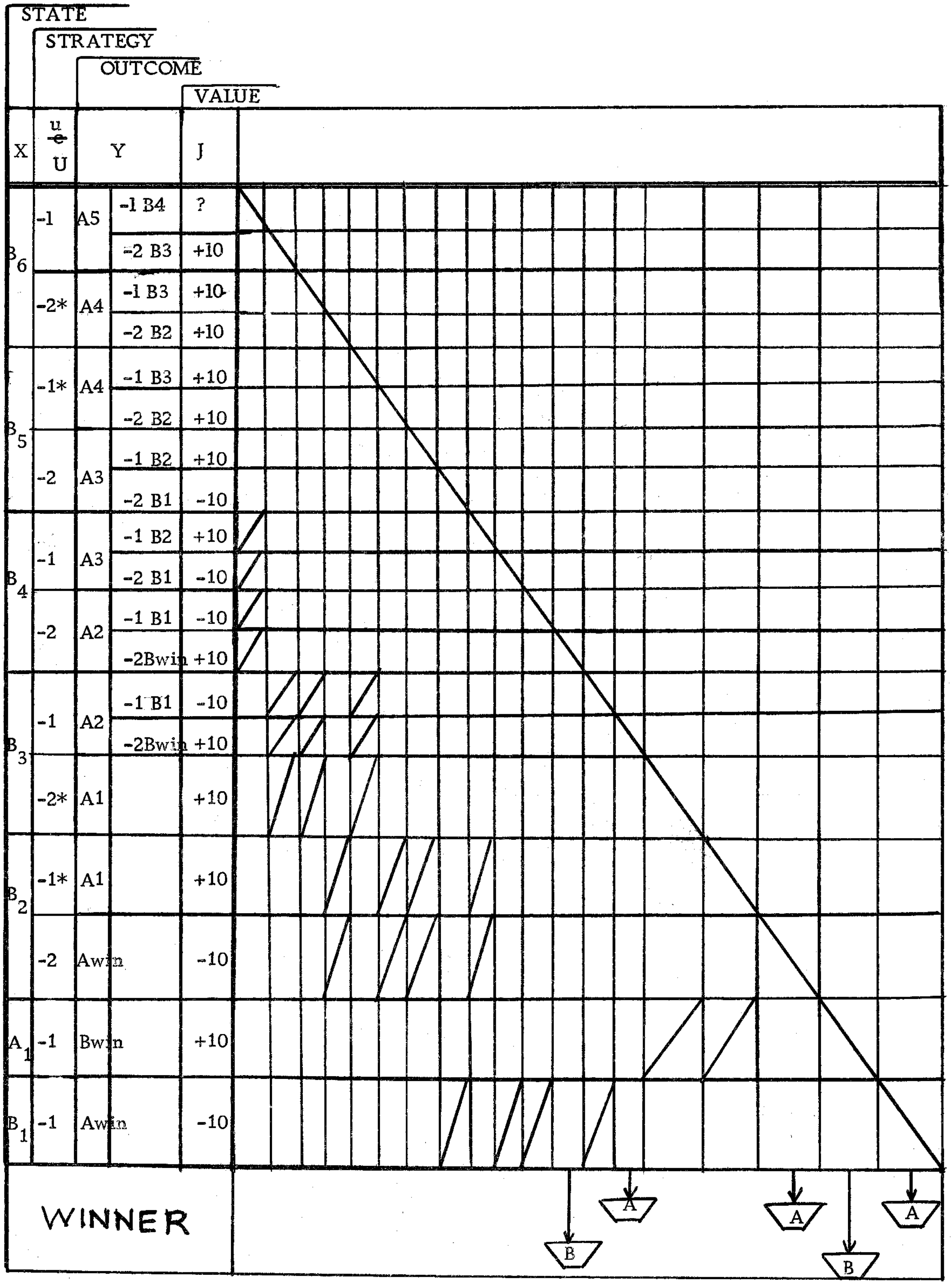


Figure 6.17. Six-match game against nature.

other hand, we win with $J = +10$. The value of the game at this stage, therefore, will be $p(-10) + (1-p)(+10)$ where p is the probability of A choosing his winning strategy. Previously, we have assumed an intelligent opponent $(p = 1)$, who is eager to win. But in practice, his objective function may not be as simple. Aside from the obvious case where he has no intelligence (say $p = 1/2$ for a random choice), it is possible that he is taking other factors into consideration such as the duration of the game. For example, each move made by him or us may be costing him \$10 per move. His objective function J will indicate that he is indifferent to either strategy: -10 in either case. By offering him 50 cents, it is very likely that you can persuade A to lose the game.

This is not unlike what happens on a negotiation table during a strike. The strike that is costing a fortune (per day) may be settled by making an almost trivial concession. Because of the pride and bias involved, we often need a third party to realize such a concession.

Zero-sum game. The case of B4 is even more interesting. By redrawing Figure 6.8 as in the cases of B3 and B4 shown below in Figure 6.18, we realize that a game matrix is essentially a snapshot of a system relating its alternatives to their outcomes.

In B4, we have the familiar zero-sum, two-person game. In order to call it a true game, we should add the constraint that A

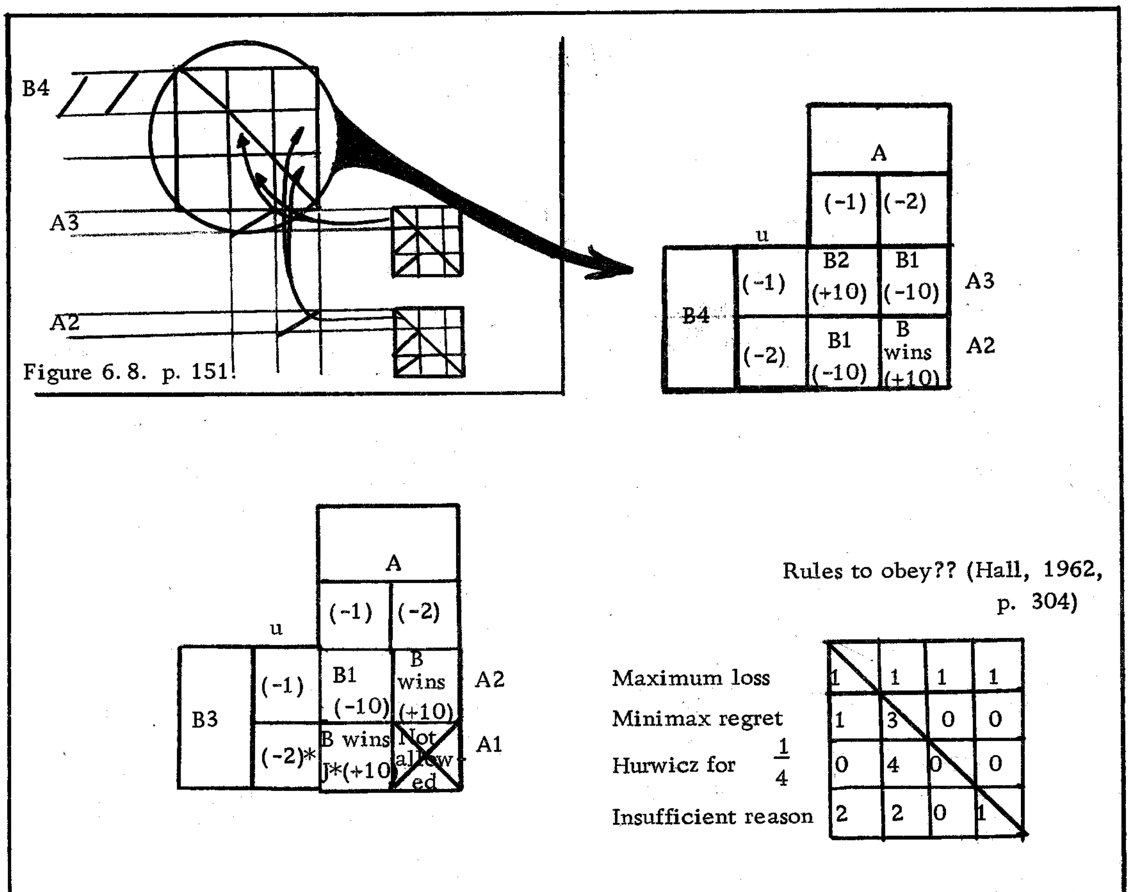


Figure 6.18. Two-person games.

should make up its mind before B shows his move. But even if this were not the case, as far as B is concerned, the game approach may be the only way to salvage any chance of winning. As long as p is not 100 percent but say 95 percent, there is definitely a chance (five percent that A will make a mistake) for B to win, especially if there are many moves to be made before the end of the game is reached.

<u>Advantages of tableau over game matrix.</u> A game matrix may be an ideal representation, when there are only two persons and

where the outcomes are readily found and are computable. However, when the game becomes complex, for example, in the case of our 24-match game, constructing a matrix for the "lose" state of B21 will not be an easy task. A tableau has the advantage that the sub-tableau at each stage may be considered as a summary and the details of any particular immediate outcome may be traced to another sub-tableau giving the information needed. This process may be continued until the very end of the game is reached (assuming a terminal game) or until a steady-state condition is reached (assuming an ergodic process).

After the desired information is obtained to fill a game matrix from the study of a tableau, all the wealth of Game Theory may be tapped to analyze the situation.

According to Dr. Melville C. Branch, the Chairman of Planning of the Los Angeles City Commission, there appeared recently an article titled "Magic Number 7 ± 2", by a renowned mathematician, discussing the limitation of human ability to manipulate no more than about seven variables with about seven alternatives.[15]

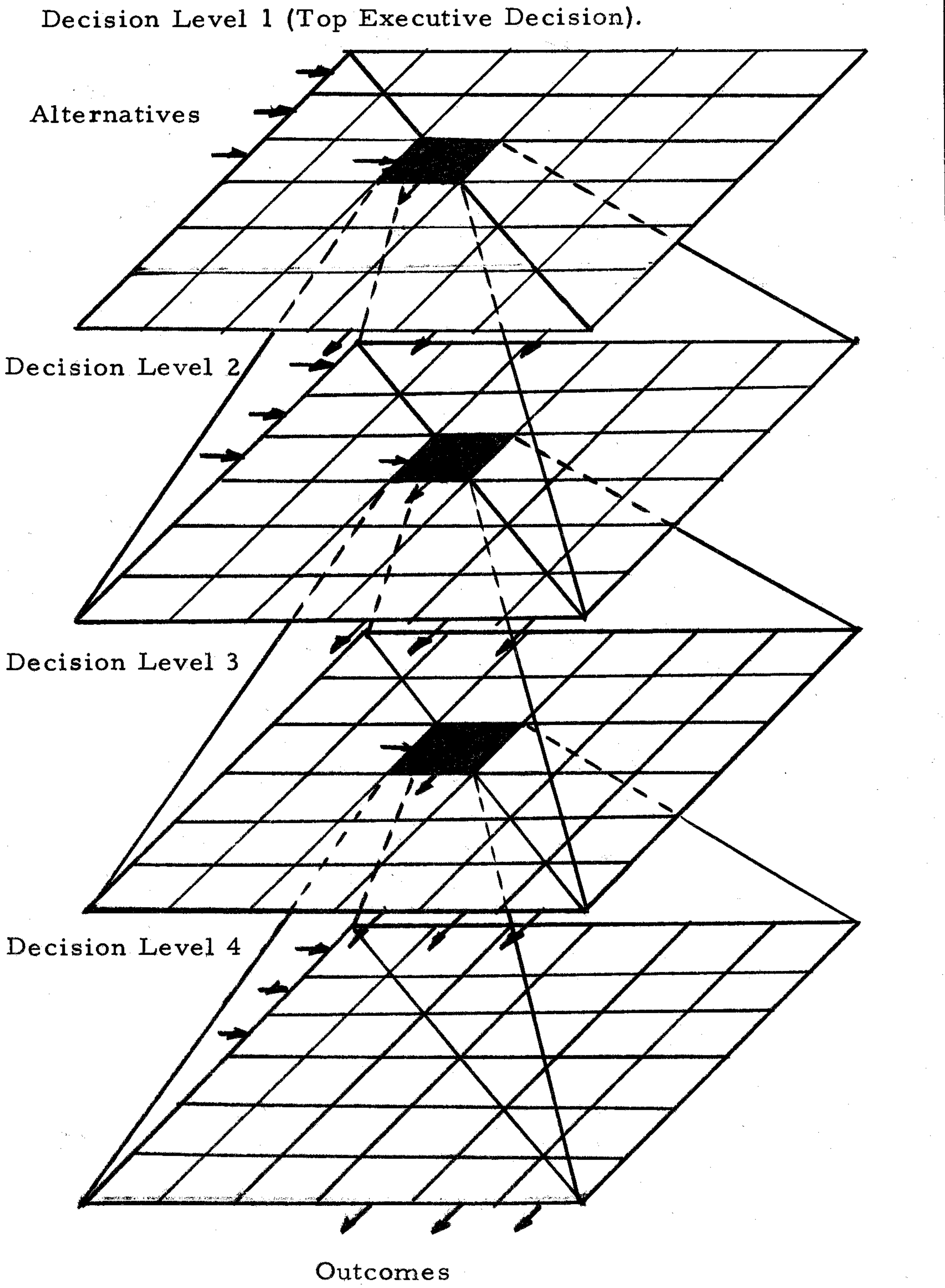
---

[15]This is another "marvel of our decimal system" in that we have chosen the limit of our mental capability as the range of digits. Twelve would have been too many for us to distinguish, less than ten, say four, would have made the expression of larger quantity too cumbersome. At any stage of our numerical computation, we are only confronted with a simple choice of ten alternatives: e. g., the unit position of two numbers being multiplied is either 0, 1, . . . , or 9, etc.

A tableau can be constructed to meet this requirement. It can be subdivided so that a decision-maker is never confronted with more than seven alternatives at a time. Each alternative's outcome might have been computed by a management level below using a sub-tableau that also confronted them with seven alternatives or so. The expected payoff for those alternatives might have come from a management level that is using a sub-sub-tableau, and so on. This process is pictorially represented in Figure 6. 19.

The top management's decision to select an alternative is, therefore, based on a weighted average of results forwarded from various sub-tableaux. Once the weighted average exceeds a certain threshold, the input (alternative) is decided. This is essentially what we will call a "Bang-bang" control: a discrete control based on more or less continuous input functions.

This discussion could lead us into Wald's consideration of decision process as a Nature vs. Management game involving stochastic considerations, the application of Bayes' theorem for estimating opponent's (Nature's) behavior, and other interesting topics such as Stochastic Programming and so on. But before we can even talk about these topics, it is necessary to discuss what Dynamic Programming is and how it can be implemented on a tableau (the subject of the next chapter). Before closing this chapter, let us consider one more topic that is closely associated with the discussion of



A Tableau-tree Concept

Figure 6.19. Tableau as a management decision-making tool.

an Automaton, namely the subject of neuron net as we have introduced in Chapter I (page 22).

Modular Network

Finite Automata and Modular net. In our formulation of Tableau Method in Chapter III, we have made our tableau representation completely interchangeable with a network representation of a system. In this chapter, we have shown the great similarity that exists between such things as Algorithm, Turing machine, Automaton, and our Tableau. In discussing how a top-management can use a tableau to initiate a Bang-bang control based on a weighted average of expected payoffs, our readers were reminded of the TLU (Threshold Logic Unit) discussed in Chapter I.

We can now conceive how our brain and our nervous system can be represented by a modular net, and every modular net interpreted as a finite automation. This concept, now considered as an essential part of Cybernetics, was initiated by two neurophysiologists: Warren S. McCulloch and Walter Pitts who first published their work in 1943 as "A logical calculus of the ideas immanent in nervous activity" in the Bulletin of Mathematical Biophysics (vol. 5).

> Mr. Pitts had the good fortune to fall under the McCulloch's influence, and the two began to work quite early on problems concerning the union of fibers by synapses into systems with given overall properties. Independently of Shannon, they had used

the technique of mathematical logic for the discussion of what were after all switching problems. They added elements which were not prominent in Shannon's earlier work, although they are certainly suggested by the ideas of Turing: the use of the time as a parameter, the consideration of nets containing cycles, and of synaptic and other delays (Wiener, 1960).

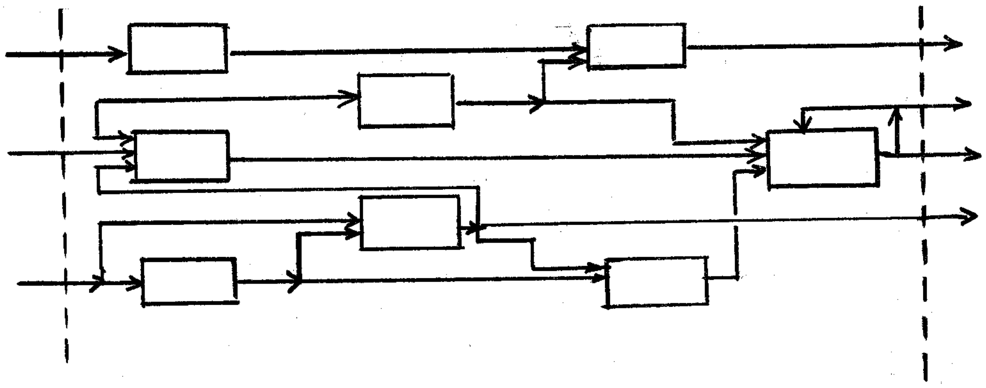


Figure 6. 20. A sample modular net.

McCulloch-Pitts model. In order to point out the exceptional ressemblance of McCulloch-Pitts model of neurophysical study to what we have been calling an automaton, and to suggest the theoretical possibility of representing this biological system by a tableau, we will have to ask our readers to be satisfied with the comparison of the definitions of a module and a modular net which we will quote below with the definition of an automaton as we have been using (page 15).

Definition 6. 4. Module (formal neuron)(Arbib, 1964).

A module is an element with, say, m inputs $x_1, \ldots, x_m$ $(m \geq 1)$ and one output d. It is characterized by m+1 numbers, its threshold θ, and the weights $w_1, \ldots,$

w_m, where w_i is associated with x_i. The module operates on a discrete time scale $t = 1, 2, 3, \ldots$ the firing of its output at time n+1 being determined by the firing of its inputs at time n according to the following rule: The module fires an impulse along its axon at time n+1 iff the total weight of the inputs stimulated at time n exceeds θ of the neuron. If we introduce the symbolism

$m(t) = 0$ for 'm does not fire at time t'

$m(t) = 1$ for 'm does fire at time t'

(where m may be an axonal output or a synaptic input of a neuron), we see that the above rule may be expressed as:

$$d(n+1) = 1 \text{ iff } \quad w_i x_i(n) \geq \theta$$

Note that a positive weight $w_i > 0$ corresponds to an excitory synapse (i.e., module input) whereas a negative weight $w_i < 0$ means that x_i is an inhibitory input.

Definition 6.5. Module net (Arbib, 1964).

A modular net is a collection of modules, each with the same time scale, interconnected by splitting the output of any module into a number of lines and connecting some or all of those to the inputs of other modules. An output may thus lead to any number of inputs, but an input may only come from at most one output.

Philosophizing. If our hypothesis is correct, and if indeed our brain can be thought of as an automaton, or an expression of algorithms, then Gödel's Incompleteness Theorem should equally apply to our thinking which makes use of our brain. If we can think of Nature as the Algorithm that created these sub-algorithms (men, animals, etc.), then we may conclude that the true understanding of the Algorithm cannot possibly come from the use of the sub-algorithms. In other words, to understand the Metamathematical Truth, we cannot use mathematics to reason. Thus, in order to understand Nature truly and completely, we cannot make use of our brain alone, but we need a Mechanism that functions at the Metamathematical level. Such a mechanism must possess both the mathematical nature (our brain) to communicate with us, and a metamathematical Nature that is of the same level as the Algorithm that created our brains and other algorithms of our level. For such a Mechanism, the Time constraint does not have to hold any more than other dimensional constraints under which we must operate. If death is our liberation from our thinking being bound to operate through our brains, and if our thinking can operate without our brains after our death, there is a possibility that we may then understand truly all of the metamathematical truth. Of course, then, it becomes impossible for us to communicate back any truth we have found: "Within any world with which we can communicate, the direction of time is uniform" (Wiener, 1960) and helas, we have then lost our measure of "time" (we are only conscious of time as a concept resulting from our observation of change in our physical system or its environment).

This is not a treatise in theology, and we do not intend to make reference to any religious truth. This discussion is presented merely to point out how far our system's philosophy could extend our thinking, and how our past "intuitive" feelings that resulted from the consideration of cosmos as a system, does agree rather closely with a more materialistic, if methodical, formulation of systems theory. Pascal once remarked "S'il n'y avait pas de Dieu, il fallait L'invinter" (If there were no God, we should invent Him). In order to give meanings to our systems, we must assume a Meta-system, so that all our systems can be considered as sub-systems obeying the incompleteness, and thus consistent nature of Gödel's theorem.

Let us close this chapter with a quote from Stafford Beer (1959, p. 97):

> Moreover, a rather more subtle logical argument shows that (since the route to a given state may have followed several circular paths) it is not possible to work out in retrospect the correspondance of machine's states to the passage of time. Hence argued McCulloch and Pitts, the human being's knowledge of the external world is necessarily incomplete. The human brain is compelled to abstract from its experience, and is incapable of becoming a machine for slavish repetition of its own reactions. Not, of course, that this is disadvantageous: the process is in fact precisely the one which enables us to systematize, codify and make use of our experience.

The concept of Homomorphism is now seen as the cause (at least to our thinking) of the irreversibility of the flow of time. We could conceivably slow down the flow of time as a dimension, but not reverse it without having to rely on some other system beside our brain. This, of course, is dependent on the assumption that McCulloch-Pitts model is a sufficiently adequate presentation of our actual brain.

VII. MULTISTAGE DECISION PROCESS

In the period following World War II, it began to be recognized that there were a large number of interesting and significant activities which could be classified as multistage decision processes. It was soon seen that the mathematical problems that arose in their study stretched the conventional confines of analysis, and required new methods for their successful treatment. The classical techniques of calculus and the calculus of variations were occasionally useful in these new areas, but were clearly limited in range and versatility, and were definitely lacking as far as furnishing numerical answers are concerned (Bellman and Dreyfus, 1962).

Dynamic Programming

Synopsis of this Chapter

This chapter is divided into three main parts. In the first section, we shall develop the tableau as a tool in solving Dynamic Programming Problems.

In the second part, we will devote our attention to a very simple but important development from the application of Dynamic Programming, namely the Utility Theory.

In the last section of this chapter, an attempt will be made to view the Control Technique field and its relationship to tableaux.

Sequential Decision Process

Present state of art. The optimization method to which the Tableau Method is particularly well suited is a multistage sequential decision process where the return function is monotonic. Without explicitly defining as such, we have already been solving problems of this nature in Chapter VI. Dynamic Programming is a term coined by Richard Bellman, supposedly while he was in search for a "dynamic" word to describe his brain child, a new development in mathematical programming (Denardo, 1966).

His book "Dynamic Programming" which appeared in 1957 from Princeton, is usually considered to be the foundation for this theory.[16] The subsequent development is mainly due to Bellman [he is ahead of others by some 200 publications (Wolfe, 1966)], but important contributions are acknowledged by Bellman as due to: S. Dreyfus, M. Aoki, T. Cartaino, M. Freimer, O. Gross, R. Howard, S. Johnson, R. Kalaba, and W. Karush (Bellman and Dreyfus, 1961). Of those, we shall only examine the works of Dreyfus, Kalaba, Howard and his colleague Kimball.

Computer programs. The major disadvantage of Dynamic Programming stems precisely from the too versatile nature of its

---

[16] The computational techniques were developed by Bellman at RAND in the early 1950's.

Principle of Optimality. This all-encompassing principle has made Dynamic Programming to be applied to such diverse problems in varied fields that a proper unified codification and standardization of algorithm has not been possible. In spite of the development of Policy Improvement procedure by Ronald Howard and his claim:

> Note that all arguments above (Policy Improvement) apply equally well to both the discrete and continuous cases. As a result a single computer program may be developed to solve both types of processes (MIT, 1959, p. 173).

no universally known standard program has been developed for Dynamic Programming.

This fact should be contrasted with the case of Linear Programming where a Simplex (or a modified Simplex) Method Program is available for practically any major commercially available computer produced in any country (USA, England, France, Japan, and Russia). There are other non-linear programming techniques which are also available today:

Quadratic Programming (RAND)

Gradient Projection Technique (Rosen)

SUMT (Sequential Unconstrained Maximization or
Minimization Techniques by Fiacco-McCormick, RAC)

Separable Programming

Others (Wolfe, 1966).

A Dynamic Programming problem, on the other hand, is usually programmed individually each time a problem is to be solved. Since writing a program is a major task even when the algorithm is known, many Dynamic Programming problems are being solved by one of the above methods for which a standard program is available.

> Consider first the one resource process. . . the simplest type of dynamic programming process. The coding of such a program for a high-speed computer. . . , can be accomplished in a couple of days using Fortran (Bellman and Dreyfus, 1961).

Concept of a "State"

Set theory. Every so often, it becomes necessary for us to review our path and to understand the new concepts in light of what we have examined in the past. Before we start our formal discussion of Tableau Application to Dynamic Programming problems, perhaps we should review how our concept of a "State" has come to exist (see Chapter III, p. 84).

Essentially a "State" is but an element of a subset. We may use the term "State Space" to determine the subset that is composed of all the elements which we consider "states." In the case of our Grammatical Tableau (p. 149), the state space Ω was composed of three elements: input, action, and output. In the case of the six-match game discussed in Chapter VI (p. 170, for example), the state space may be defined as:

$$\Omega = \{\text{Start, A6, B6, A5, B5, A4, B4, A3, B3, A2, B2, A1, B1, Awin, Bwin}\}$$

In the case of Ashby's Ghost (p. 178) we had:

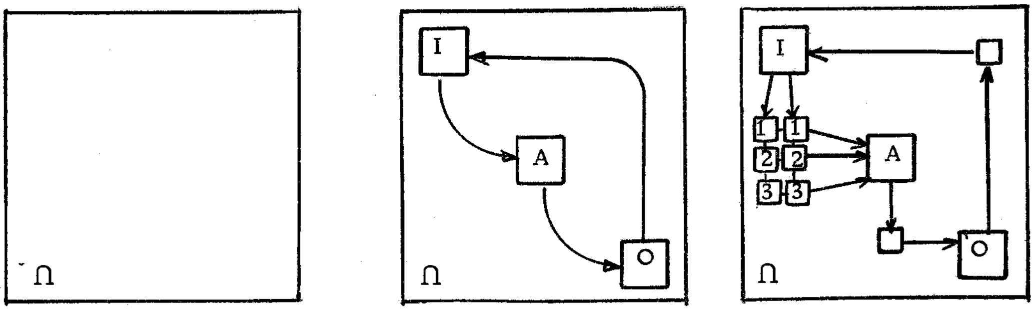
$$\Omega = \{11, 10, 01, 00\}$$

In each case, our knowledge of a particular state was confined to "how to get there" (controllable input), which told us how a state could be mapped into this element, and "what happens then?" (observable output), which told us the outcomes to which this element can be mapped to. This "black box" concept makes us realize that a state is an agent relating two subsets, an independent subset which we call "inputs" and a dependent subset which we call "outputs." The mapping relationship with the first subgroup was called "fan-in," while the mapping relationship with the latter, "fan-out." The fan-out was usually determined by Nature (we usually had no control: Ghosts in Ashby, Opponent in six-match game) while the fan-in relation was "decided" by the management from a set of possible alternatives. Of course, there have been systems where this fan-in was not controlled by the management but by Nature such as the random choice according to an a priori probability in our "Grammatical Tableau," or by a truth known only to Nature as in the case of the twelve-ball game (p. 160).

In the Dynamic Programming applications, we are especially interested in cases where these decisions are left to the Management.

Dynamic Programming tells us which alternative should be chosen to yield the highest utility for the overall system.

Figure 7.1 below shows the State Space for our Grammatical Tableau example.



State Space Ω

Ω = {I, A, O}
States

Decisions Alternatives 1, 2, 3

Figure 7.1. State Space for Grammatical Tableau.

<u>Control theory</u>. While mathematicians are concerned with the relationships between the state as an element and other elements constituting its environment, control engineers would like to call these alternatives in mapping, "inputs" or "controls." A computer programmer would prefer the term "conditional branches," while an Operations Researcher would prefer the word "decisions." With some generalization, we observe that "the law of jargon" (p. 41) prevails here too. The less a system is understood, and the less freedom is left to make a choice, the more ostentatious the word becomes to describe these inputs. A "management decision" will be based on "experiences" of managers who listen to their OR men.

It becomes an "executive decision" when irreducible factors are taken into considerations by the top management which listens to its middle-managers' management decisions. The pell-mell of all the past executive decisions becomes a bible called "corporate policy" which should be consulted for future decisions. When the society becomes used to decisions based on these policies, it starts to call them "social ethics," "right to . . . ," and other unwritten rules.

If there is, however, one common feature that everybody agrees, that is the grave "consequences" that may result to the "state" of the system depending on the input made. Nobody really knows what the "state" of the system is, but everybody is certain that the input will have some effect on it. Thus, when the necessity arises to describe the system, instead, we give an account of the inputs to affect its state.

> A fundamental notion is that of the state of a system. The prior inputs that a system has experienced will generally affect its state (or condition of being) at any instant. That is, the net effect of all past inputs may be summarized by specifying the state of the system at that instant. Two different prior input patterns that result in the same state of the system at a given instant may therefore be considered to be identical insofar as the future evolution of the system is concerned (Huggins; Flagle et al., 1960, p. 668).

Multi-stage decision process. When the tableau is empty, it may represent a state space but will not provide adequate descriptions about the states themselves. A state can be adequately described when we know either the entry into its own eigen-cell or all the transducers feeding into and from it. Thus, a transducer may be considered as an information channel that serves to describe a state. Perhaps, an adequate definition of a "state" is the one given by Denardo (1966): "A state is a synopsis of history sufficient for costing future actions." A more complete one is given by Kimball and Howard:

> In any system undergoing a process of the multistage decision type, certain information is needed at each decision point in order to make that decision... The term state will be used to denote all the information needed to describe the system at any stage. The stage is usually described in terms of the values of the set of variables. In some cases, these variables are discrete, in others they are continuous. As the process we are considering proceeds, the state of the system is ordinarily constantly changing (Kimball and Howard, MIT, 1959).

Multi-stage Decision Problem

Decisión rule. The problem that we deal under the name of multi-stage decision process, is the one that includes a discrete number of decision points, states at which management has control of alternatives. The decision to select the particular alternative, is essentially the problem of selecting the control input $u_1 \in U$ from the subset U of transducers available at that state.

The dilemma is presented because each alternative has a different effect on the state. The solution to this dilemma is to choose an alternative or a weighted combination of the available inputs. In order to yield an objective decision, each alternative's total benefit (profit) must be shown in a numerical value, i.e. the objective function must really be a functional that will yield a numerical evaluation of each alternative that can be compared against a threshold value θ, a number in the same measure.

Boundary conditions. The system is supposed to be in a given state $\underline{x}_o = \underline{x}(t_o)$ at the beginning: $t = t_o$. This initial state is usually described and given, though in some cases (as we shall see in ergodic cases) it is neither given or needed.

A decision process may be terminated in three ways:

1. When time t reaches a certain predetermined terminal time t_1. For example, $t_1 = 5$ years ($t_o = 0$ automatically assumed).

2. When a terminal state is reached by the system. $x(t_1)$ is given. For example, $x(t_1) = 50,000$ miles.

3. A combination of both, or a function of both. $T[t_1, x(t_1)] = 0$, a terminal condition, is specified. For example, a guarantee on a car that lasts five years or 50,000 miles, repair or replacement to the option of the manufacturer (a sequential decision process if the car breaks down often).

Another interpretation of this termination is the concept of limited reward: "The simplest multistage decision processes are those in which a state is reached such that the rewards stop when that state is reached" (Ronald Howard: MIT, 1959, p. 157).

Allocation Problem

Example Problem 7. 1. Let us consider a simple allocation problem in which we are to distribute a limited amount of resource, say 1, 000, 000 dollars, to six projects, 1, 2 . . . 6.

Our aim is to maximize our total satisfaction. Obviously, this is easier said than done. Each project has a certain amount of risk involved with it, and the decision-maker (the player, manager, etc.) must be able to attribute a functional[17] that will yield a numerical "value" for each project taking into account these risks. Formulating each project as a separate identity is itself a job, to attribute a common numerical scale of utility is almost an impossibility, but to assume the existence of a common base for comparison is certainly outrageous. Nonetheless the modern utility theory does give us some help toward this direction:

> In brief, the current theory shows that if one admits the possibility of risky outcomes, i. e., lotteries involving the basic alternatives, and if a person's preferences are consistent in a manner to be prescribed, then his preferences can be represented numerically by what is called a utility function. This utility has the very important

[17] "A transformation from a function to a number is called a functional: a transformation from a function to a function is called an operation" (Bellman and Kalaba, 1965, p. 5).

property that a person will prefer one lottery to another iff the expected utility of the former is larger than the expected utility of the latter. Thus, the assumed individual desire for the preferred outcomes, becomes in game theory, a problem of maximizing expected utility (Luce and Raiffa, 1957, p. 4).

Classical approach. Let us suppose, for convenience, that we have managed to arrive at the utility functions for the six projects to which the fund of $1000G is to be allocated.

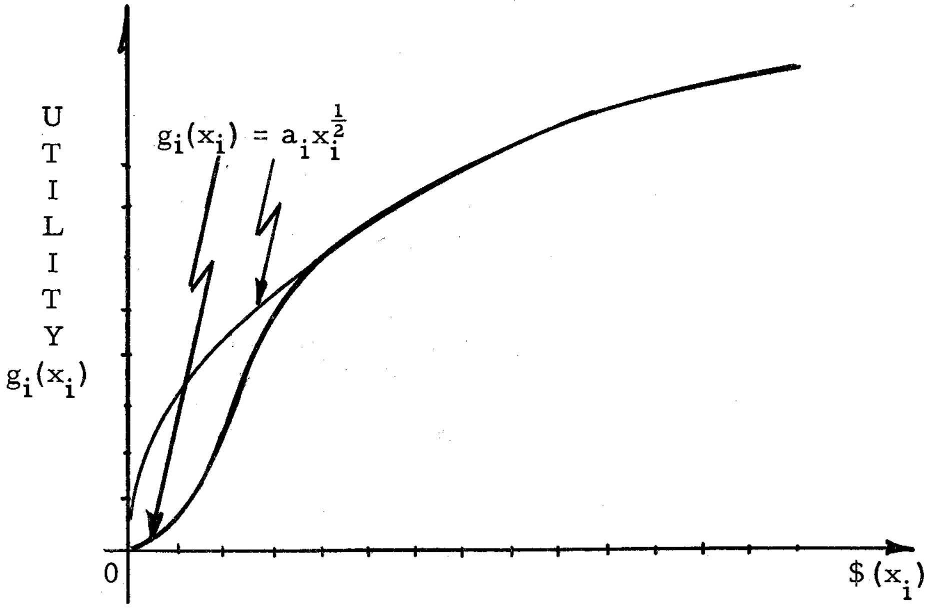


Figure 7. 2. An approximation to a utility function.

Moreover, let us assume that we have conveniently found that each utility function can be approximated by a positive square root function as shown in Figure 7. 2. In other words, we are assuming that the operation (with $1M) will be in the region of diminishing return for each project.

Another blunt assumption we make, is that the utilities obtained from the various projects are additive. Later discussion on utility will clarify some of these assumptions (page 233).

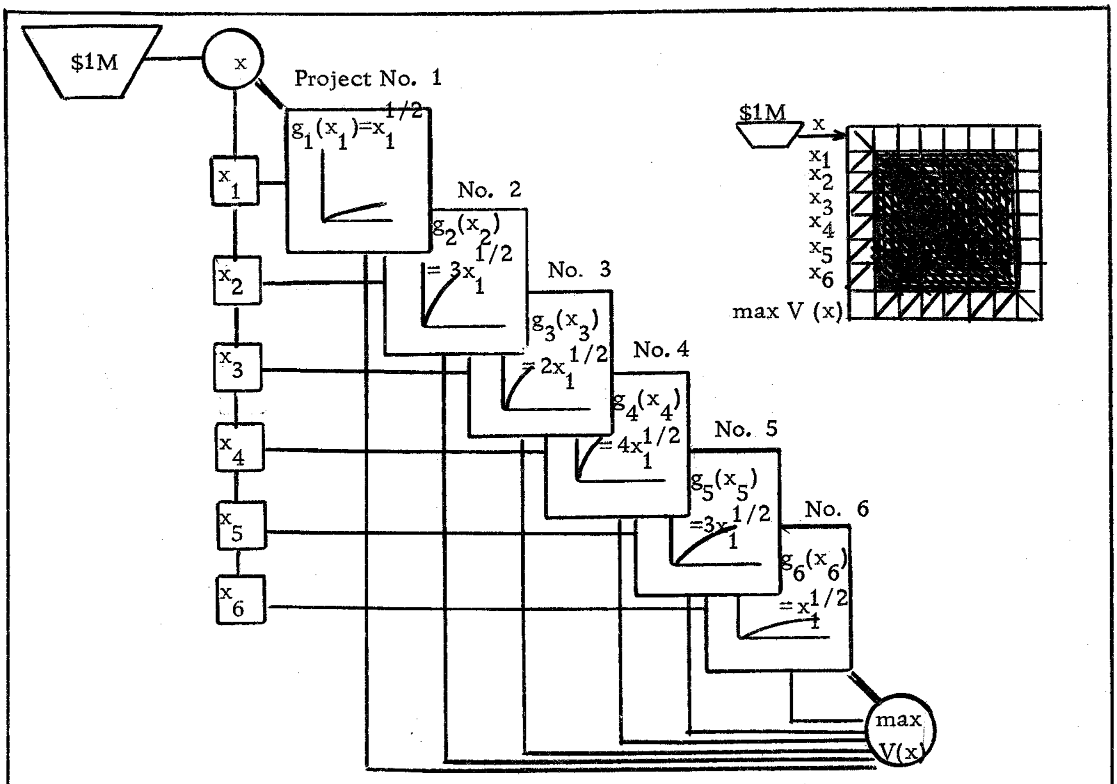


Figure 7.3. Tableau interpretation of the Problem 7.1.

Thus, the problem is essentially:

$$(\text{maximize})V(x) = g_1(x_1) + g_2(x_2) + g_3(x_3) + g_4(x_4) + g_5(x_5) + g_6(x_6)$$

$$= x_1^{\frac{1}{2}} + 3x_2^{\frac{1}{2}} + 2x_3^{\frac{1}{2}} + 4x_4^{\frac{1}{2}} + 3x_5^{\frac{1}{2}} + x_1^{\frac{1}{2}} = \sum_{i=1}^{6} a_i x_i^{\frac{1}{2}}$$

subject to the constraint:

$$x = \sum_{i=1}^{6} x_i = 1,000,000 = 1000G$$ (we shall conduct our computations in G's rather than in $'s).

We should not forget that our analysis is possible thanks to the celebrated Weierstrass' theorem of extremum.

Theorem 7.1. Weierstrass' Theorem. Every function which is continuous in a closed domain G of the variables possesses a largest and a smallest value in the interior or on the boundary of the domain (cf. Courant and Hilbert, 1953, p. 164, etc.).

The readers may realize that we have been somewhat sloppy in our use of Lagrange Multiplier method to prove the maximum information theorem (p. 158), in that we have not checked for our boundary conditions. The criticism is legitimate but an intuitive justification was given in the form of Figure 6.2 (p. 160) showing that the extremum we had found was indeed the maximum, minimum occurring at the boundary.

We can take the same method to solve our problem:

Our function F becomes:

$$F = \sum_{i=1}^{6} g_i(x_i) - \lambda \sum_{i=1}^{6} x_i \qquad \text{where } g_i = a_i x_i^{\frac{1}{2}}$$

and therefore: $\lambda = g_i'(x_i)_2 = \frac{1}{2} a_i x_i^{-\frac{1}{2}}$ for $i = 1, \ldots, 6$.

$$\lambda = \frac{a_i}{a(x_i)^{\frac{1}{2}}} \quad \text{or } x_i = \frac{a_i}{4\lambda^2}$$

thus, $$x = \sum_{i=1}^{6} x_i = \sum_{i=1}^{6} a_i^2/4\lambda^2 = 1000G$$

and $$\lambda^2 = \frac{\sum_{i=1}^{6} a_i^2}{4x} = \frac{40}{4000} = \frac{1}{100} \qquad \text{or } \lambda = 0.1$$

that is if: $a_1 = 1$, $a_2 = 3$, $a_3 = 2$, $a_4 = 4$, $a_5 = 3$, and $a_6 = 1$.

The budget can be obtained from:

$$x_i = \left(\frac{a_i}{2\lambda}\right)^2 = 25\, a_i^2 \text{ (in G's)}$$

The budget is shown below, and the maximum satisfaction is about 6324.6, compared to 4000.0 if the manager invested all funds into the highest gain investment: project no. 4.

Budget:

Project no. 1:	\$ 25,000	
Project no. 2:	225,000	
Project no. 3:	100,000	
Project no. 4:	400,000	
Project no. 5:	225,000	
Project no. 6:	25,000	
Total Budget		\$1,000,000

Figure 7.4. Budget result for Example 7.1.

Approximations of utility functions. If we were to simulate the lower portion of the utility function, an expression of the type $g_i(x_i) = a_i x^2$. If such functions were used, our formulation would have yielded minimum rather than maximum at the value of λ. The maximum would have occurred at the boundary, suggesting that the investment should be concentrated to one project with the largest a_i.

The morale of the story is: "If you are rich, diversify: if you are poor, specialize."

Between the minimum and the maximum problems for the Lagrange Multiplier, there is of course a problem that will yield an infinite number of solutions. This is the case where the power of x becomes unity, or where g_i will be a linear function of x_i. The manager will be "indifferent" to the choice of satisfaction/projects levels as the ratio will then be a constant. When we actually face such a degeneracy, the solution most usually adopted is to perturb our utility function by adding some so-called irreducible factors such as "customer image," "social responsibilities," "safeguard against future government interference," or "labor good-will," to render the function non-linear (ever so slight) and to settle on one solution.

Another very powerful approximation of a utility function is the use of logarithmic expression. We have a rather interesting interpretation of this particular topic, but we would like to postpone this discussion until the discussion of "Efficient Gambler" (Problem 7.4) where we shall actually make use of a logarithmic utility function.

Classical vs. Dynamic Programming approach. The rather naive example of 7. 1 does point out several features of the classical analysis technique. First of all, the amount of efforts and assumptions necessary in obtaining suitable data is overwhelming. Asking a manager to give an ordinal measure of utility (e. g. no. 4 is better than no. 5, etc.) is already a difficult task, but to demand a cardinal measure (e. g. how much of no. 4 for so much of no. 5, etc.) that is continuous to satisfy Weierstrass' theorem, is an impossible assignment. The utility function should not have any discontinuity and must be monotonically increasing. A point of inflection would give a false indication of an extremum. Also, we have the annoying task of checking for boundaries. Bellman likes to give heated arguments against the classical methods (as it is obvious from almost any of his writing) and most of them are valid. They were useful in promoting his Dynamic Programming when it was relatively unknown. But now that Bellman's principle of optimality is found to be closely related to the classical calculus of variations (see discussion later in this chapter; Bellman and Dreyfus, 1962, p. 180; or Leitmann, 1962, p. 255) it seems that more efforts should be directed toward sharing their strengths than to attack their weaknesses. We are here to show the applicability of Tableau method and not to condemn any particular method. Solving Problem 7. 1 by Dynamic Programming would indeed be a much slower task.

Principle of optimality (Bellman, 1956). An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Basic optimality functional. The basic idea of the principle of optimality is the same as the one involved in a tournament or in a contest: the champion will be a champion regardless of where he comes from originally and how the grouping is organized up to where he is recognized as a champion. Counties may elect their champions, states from county champions, and the nation from the state champions, and the Olympic champions will be selected from the athletes representing nations. In each case, the decision should be the optimal with regard to the champions represented in the particular tournament, and should be a part of the optimal policy that would eventually select the world's champion.

A more formal derivation of the principle of optimality's basic functional follows.

Given a value function:

$$V_N(x) = \max[\, g_N(x_N) + g_{N-1}(x_{N-1}) + \ldots + g_2(x_2) + g_1(x_1)]$$

over the region $x_i > 0$, $\sum_{i=1}^{N} x_i = x$, where x may assume any positive value and N is any positive integer, we may split the region x into two parts, x_N and $x-x_N$; then, the maximum should be held maximum regardless of how they were split as long as the maximum is taken over both regions.

$$V_N(x) = \underset{\substack{x_1 + \ldots + x_N = x \\ x_i \quad 0}}{\text{maximum}} \left[g_N(x_N) + g_{N-1}(x_{N-1}) + \ldots + g_1(x_1) \right]$$

$$= \underset{0 \leqslant x_N \leqslant x}{\text{maximum}} \left[\underset{x_1 + \ldots + x_{N-1} = x - x_N}{\text{maximum}} \left(g_N(x_N) + g_{N-1}(x_{N-1}) + \ldots + g_1(x_1) \right) \right]$$

$$= \underset{0 \leqslant x_N \leqslant x}{\text{maximum}} \left[g_N(x_N) + \underset{\substack{x_1 + \ldots + x_{N-1} = x - x_N \\ x_i \quad 0}}{\text{maximum}} \left(g_{N-1}(x_N - 1) + \ldots + g_1(x_1) \right) \right]$$

$$= \underset{0 \leqslant x_N \leqslant x}{\text{maximum}} \left[g_N(x_N) + V_{N-1}(x - x_N) \right]$$

The latter is called the basic functional of the principle of optimality.

<u>Causality principle</u>. The principle of optimality may be considered as an application of the principle of causality. While the principle of causality deals with multistage processes the principle of optimatility deals with multistage decision processes. To clarify this relationship, we cannot think of better explanation than to present the direct quotation from the authorities in Dynamic Programming:

> The equations above [(talking about various multistage processes)] are all particular examples of relations obtained from the principle of causality, or, equivalently, of determinism. Let the state of a system at time t be represented by f(c, t), where c is the state at time t = 0.
>
> We can think of the system as starting in state c at t = 0 and evolving for a time s + t, in which case its terminal state will be f(c, s + t), or we can think of it as starting in state c at time t = 0, evolving for a time t, in which case its new state will be f(c, t), and then continuing for an additional time s, in which case its terminal state will be f[f(c, t), s] . . .

> Determinism asserts that both procedures lead to the same terminal state. Hence, we have the fundamental functional equation $f(c, s+t) = f[f(c, s), t]$ [(Figure 7.5)]. From this many further results can be obtained...

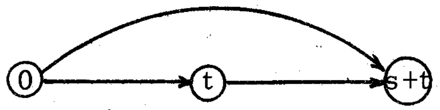


Figure 7.5. Bellman and Kalaba's explanation of principle of causality.

(Bellman and Kalaba, 1965, p. 20).

Optimality interpretation of Example 7.1. Let us view our allocation Example 7.1 in light of this basic functional of our newly found Principle of Optimality. In a nutshell, it means that if the budget as set out in Figure 7.4 (p. 218) is truly optimum, then taking any project N (say no. 6) away from the available project and its $g_N(x_N)$ from the fund (1, 000, 000-25, 000=975, 000) will still yield an optimal budget with respect to the remaining $x-x_N$ (975, 000).

This is obviously true since our formula is applicable regardless of the size of x, and in fact the same results will be obtained by using x = 975, 000 and only the first five projects.

We could use the allocation procedure to show the Dynamic Programming procedure, but it is far easier to present other examples which are simpler in structure and thus easier to follow.[18]

---

[18] Allocation problems are used as introductory problems in many dynamic programming texts (Bellman, 1957> Sasieni, Yaspan, and Friedman, 1959> Hadley, 1964).

Simpler allocation problems, with discrete return table, can be solved intuitively by considering marginal profits.

In contrast to the classical approach where all project costs were computed at once, the dynamic programming takes the approach of allocating the budget to one project at a time, using the basic objective functional to determine which would be the optimal budget at each stage.

> An essential difference between the two methods is that the recursive (dynamic programming) approach changes one problem in n variables into n problems, each in one variable. In more complicated examples, the simultaneous equations resulting from the classical calculus approach may be extremely difficult to solve; and, if more than one solution exists, we must ascertain which solution yields the absolute maximum. Worse still, the calculus approach will not necessarily reveal the maximum, subject to constraints, if it lies on the boundary of the admissible region or if we are dealing with non-differentiable functions (Sasieni, Yaspan, Friedman, 1959, p. 274).

In the examples to follow, we shall try to take as much of the "veil of mystery" away from our computational procedure by exposing as much of the details as possible. As it is becoming increasingly obvious, we have already been using the Principle of Optimality in solving problems in the last chapter. We shall adopt the same format (state, alternatives, output) in the expanded form as much as possible, resorting to the reduced form (as we did with the 24-match game) only when it is necessary. However, it should be noticed that the tableau is amenable to computations carried out directly on reduced form (as the computations have been done using CPS Tableaux), the only disadvantage being that the computations are harder to follow until the algorithms have become familiar. On expanded form, they are usually obvious.

Itinerary Problem

Example Problem 7. 2. Fairy-tale version. An Oregon State University engineering student has decided to elope with the daughter of a Corvallis millionaire (supposing that there is one). He has mapped out all the different routes by which he can drive from Corvallis to Reno, Nevada, where he plans to get married. Each route has some advantage of being faster, more pleasant, and more or less conspicuous to the State Patrol. Since the girl is still a minor, and her father is suspecting of their plot, main highways offer greater risks per mile than less frequented paths for the couple of becoming arrested by the informed State Patrol or paid detective. Which road should he take to Reno, if his reward of reaching Reno is evaluated at 30 utils, while that of risking the chance of being caught on each road is as shown on Figure 7. 6?

Example Problem 7. 2. Industrial version (originally from Denardo, 1966). A small manufacturing plant has three departments: Turning, Boring, and Shaping. In the first department, we have the choice of using engine lathe, turret lathe, or automatic machine lathe.

In the second, either a boring mill or a drill press may be used. In the third, the choice is between a milling machine or a shaper. A product will usually go through the three departments in the order above, and will incur the expense at each machine as shown in Figure 7. 6. The final product is sold at the price of $30 a piece. The expense of machine includes the discount that may be made on the final product (say $26 instead of $30) because of the lower quality of the particular machining operation.

Solution to Problem 7. 2. The seemingly unrelated problems mentioned above, are essentially identical. They are both made into the non-probabilistic dynamic programming problem by giving data in terms of expected values ($g_{ji}^{k} = p_{ji} r_{ji}^{k}$) rather than returns ($r_{ji}^{k}$). However, to keep our format uniform throughout our presentation in this chapter, we shall introduce a probability factor p_{ji} of going from state i to j. For the time being, we shall assume all p_{ji}'s to be one. The column headings are in Polish notation so that: $p_{ji} r_{ji} V_j + *$ means $(r_{ji} + V_j)*p_{ji}$ and so on (Chapter III, p. 91). i indicates the state, k the alternatives, and j the outcomes or the states to which the choice of alternative will lead to. We shall use the Diamond to mark the states which act as decision points (except the last state which is an endogenous output and which will also be marked by a Diamond: the EDP's conditional branch).

The flowgraph is shown in Figure 7. 6.

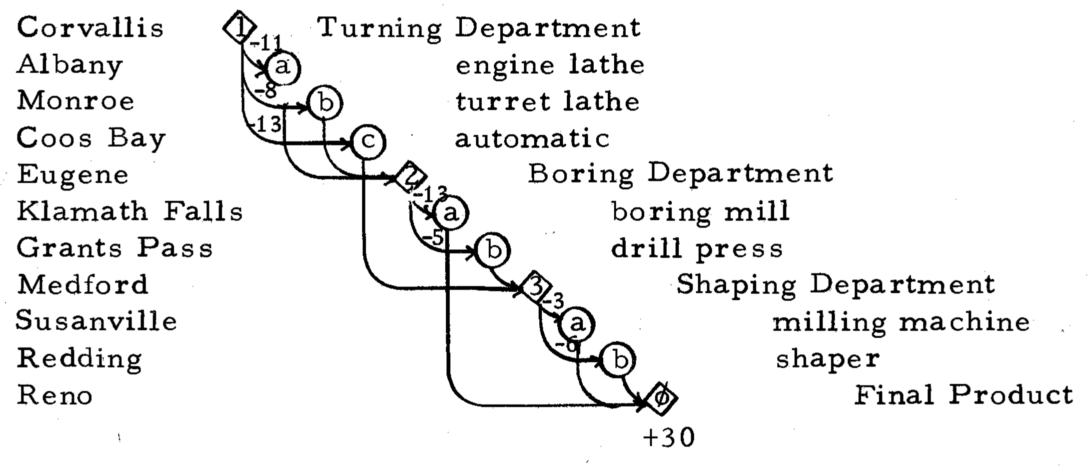


Figure 7. 6. Example Problem 7. 2 flowgraph.

The corresponding tableau is shown on Figure 7. 8. The calculation could also be carried out directly on the tableau as shown below in Figure 7. 8, but Figure 7. 7 is perhaps easier to follow at first. The computation starts at the bottom where the terminal condition is given as +30 for the empty state (terminal or endogenous output) ϕ: $r(\phi) = 30$. Using this as the V_ϕ, we can compute upward by using the basic functional, which can be written as:

$$V_i = \max_k [\sum_j p_{ji}^k (r_{ji}^k + V_j)] = \max_k (V_i^k) = \max_k [\sum_j p_{ji}^k (R_{ji}^k)]$$

This is the same basic functional as the one we have derived on page 222, except that we have added the transition probability. Note that the notation p_{ji} is the transpose of the usual p_{ij} used in so-called Markov matrix. The advantage of our notation, aside from being consistent of its location in the tableau, becomes clear when the

State i	Alt. k	Out. j	g^k_{ji}: p_{ji}	r^k_{ji}	V_j	R^k_{ji}: +	*	V^k_i: Σ_j	V_i: Max.
1									14
	a	2	1	-11	22	11	11	11	
	b	2	1	- 8	22	14	14	14	*
	c	3	1	-13	27	14	14	14	*
2									22
	a	ϕ	1	-13	30	17	17	17	
	b	3	1	- 5	27	22	22	22	*
3									27
	a	ϕ	1	- 3	30	27	27	27	*
	b	ϕ	1	- 6	30	24	24	24	
ϕ			1	+30	0	30	30		30

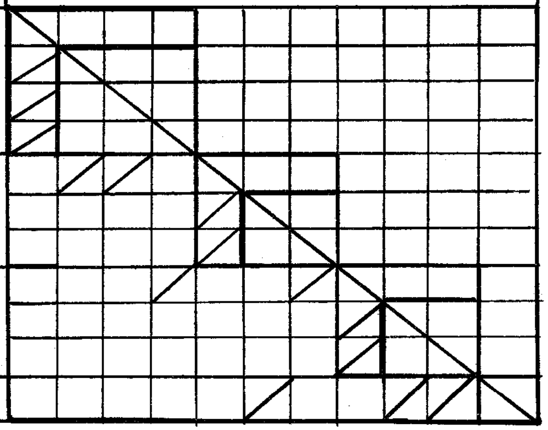
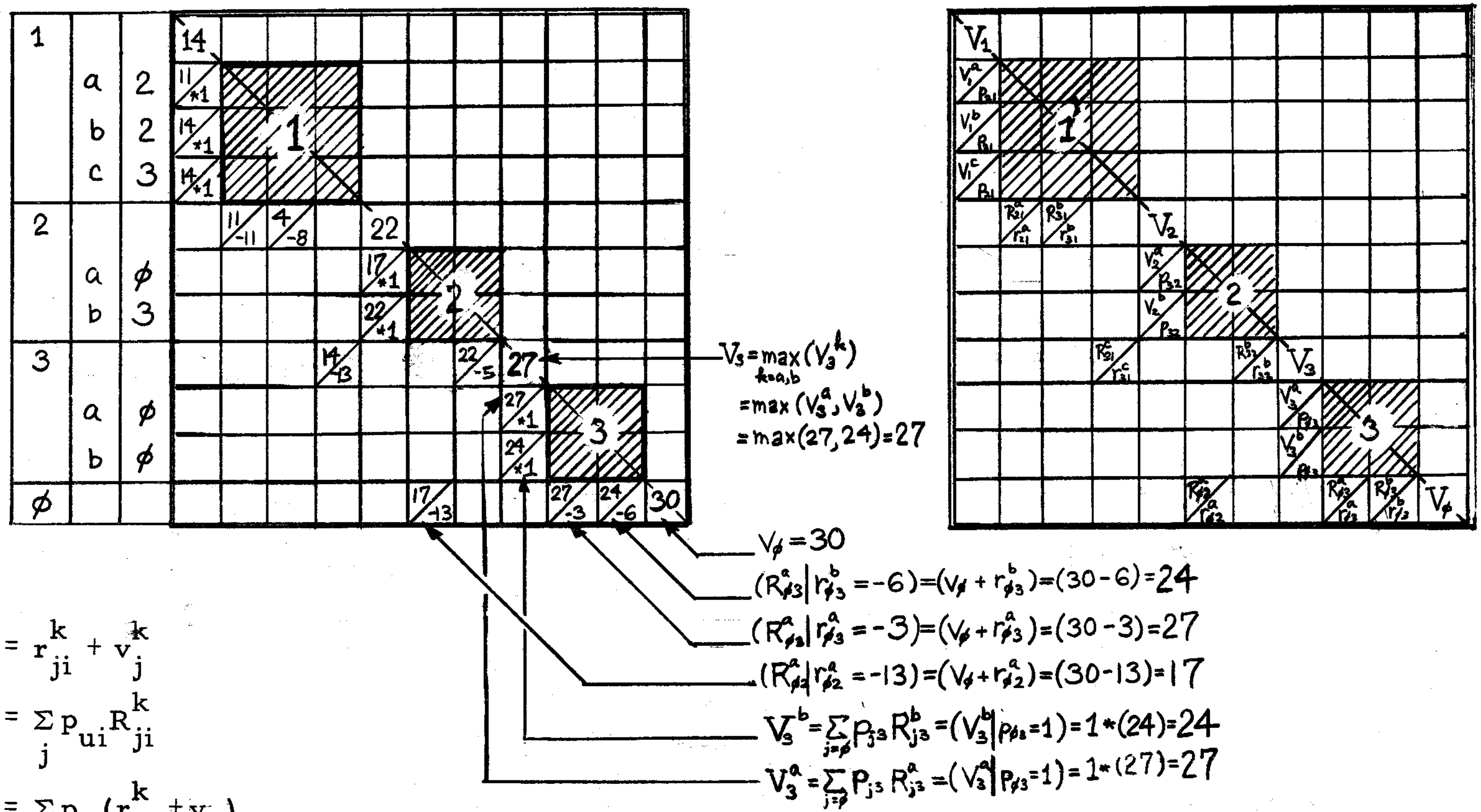


Figure 7.7. Tableau solution to the itinerary problem.



$$R^h_{ji} = r^k_{ji} + v^k_j$$

$$V^k_i = \sum_j p_{ui} R^k_{ji}$$

$$= \sum_j p_{ji}(r^k_{ji} + v_j)$$

$$V_i = \max_k (V^k_i)$$

$$= \max_k [p_{ji}(r^k_{ji} + v_j)]$$

Figure 7.8. Internal flow of data in a tableau.

Markov matrix is to be multiplied by a column vector which usually represents the state of a Markov chain.

The notations:

$$R_{ji}^{k} = r_{ji}^{k} + V_j \text{ (Polish notation +)}$$

and

$$V_i^k = \sum_j p_{ji} R_{ji}^k = \sum_j p_{ji}(r_{ji}^k + V_j) \quad \text{(Polish notation } \sum_j \text{)}$$

are introduced for bookkeeping sake. Our Polish notation makes them superficial as far as the computation is concerned but we cannot talk about + and $\sum_j$ without causing some confusion.

When the basic functional is applied to the last department (no. 3), then we have:

$$V_i = \max_k \left[\sum_j p_{ji}^k (r_{ji}^k + V_j) \right]$$

becomes

$$V_3 = \underset{k \in \{a, b\}}{\text{maximum}} \left[\sum_{j=\phi} p_{j3}^k (r_{j3}^k + V_j) \right]$$

or

$$V_3 = \underset{k \in \{a, b\}}{\text{maximum}} \left[p_{\phi 3}^k (r_{\phi 3}^k + V_\phi) \right]$$

$$= \max \left[p_{\phi 3}^a (r_{\phi 3}^a + V_\phi),\ p_{\phi 3}^b (r_{\phi 3}^b + V_\phi) \right]$$

$$= \max[1*(-3+30),\ 1*(-6+30)] = \max(27, 24) = 27$$

The process is recursive and will then be repeated for Department 2 and Department 1.

$$V_2 = \max[1*(-13+27),\ 1*(-5+27)] = 22$$

$$V_1 = \max[1*(-11+22),\ 1*(-8+22),\ 1*(-13+27)] = 14$$

This explanation may seem extremely detailed at this stage where the problem can be solved almost intuitively, but it is far easier to see the procedure when the problem is simple than after it has become cumbersome.

Reduced form of Problem 7.2 tableau. The computation required could have been carried out using a reduced tableau, or could be put into a reduced form before being presented to a higher level management. In such a case, only the optimal transducers (alternatives) will be shown on the tableau. For example, between state 1 and state 2, the transducer 1b will be selected and shown as (14/1b) meaning that the conditional transfer 1b will produce an expected return of 14. Similarly between state 1 and state 3 we have (14/1c); between 2 and 3 (22/2b); between 2 and ϕ (17/2a) and so on.

Of these transducers which are optimal with respect to the particular transition between the state i to state j, we use * to indicate that will form a part of an optimal policy. Thus, 2b* is a part of the optimal policy 1-1b*-2-2b*-3-3a*-ϕ, but not 2a, and so on. Both 1b* and 1c* are acceptable, and the top management is happy to be able to exercise their irreducible criteria to determine which alternative to choose. Figure 7.9 below shows the reduced form.

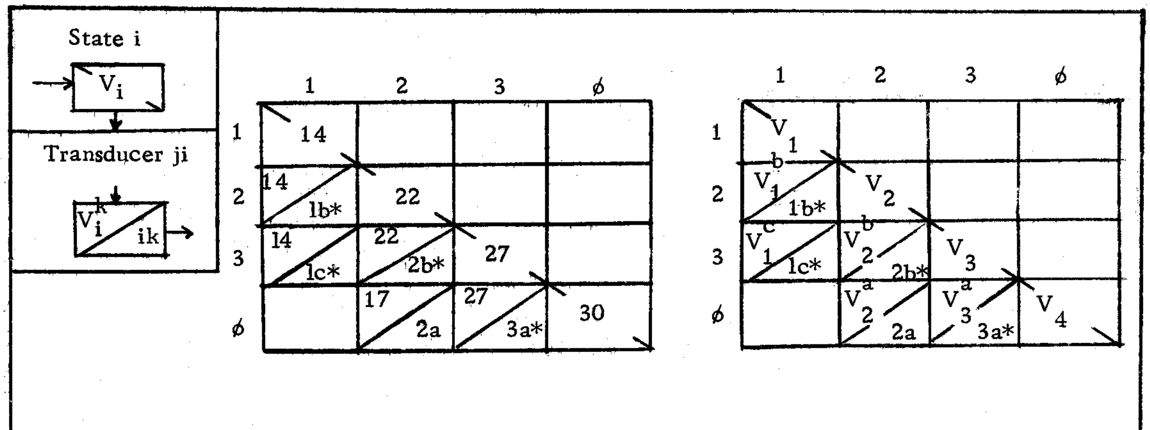


Figure 7.9. Reduced tableau for Problem 7.2.

In contrasting this with Figure 6.9, we notice that the reduced form has all eigen-cells filled. Or, more exactly, the problem is solved when we have found the entries into the eigen-cells of the reduced tableau.

The optimal policy is obvious but we shall mention it for the sake of completeness. The couple should take either Corvallis-Monroe-Eugene-Grants Pass-Medford-Susanville-Reno or Corvallis-Coos Bay-Medford-Susanville-Reno for the total utils of 14 (since its positive, it's worth eloping).

The manager should have the product manufactured by using either Turret-press-milling machine combination, or just automatic-milling machine combination. In either case he can expect \$14 profit per piece. It is interesting to note that essentially he has the choice of either making it cheap and quick, or more elaborate and careful. Obviously, other irreducibles (such as customer relations) must come into play before management decision should be made.

Utility Theory

Threshold Problem

Example Problem 7.3. Fairy-tale version. Mary is the fairest maiden in a little village. When she became of age and decided to settle down, she made a careful survey of all the young eligible bachelors in the village and selected three most eligible bachelors she would be willing to marry. She has already been asked to marry from all three boys, but is not quite sure which one she would be most happy with. She decided to get engaged to find out more about the fellows, but obviously this is possible only if she selects one at a time. She may become engaged to the first fellow for one month and decide whether or not to marry him. If he is not of the caliber she expected in the beginning, she may break the engagement and proceed with the second fellow, and so on with the third fellow. However, the fellow that she would have broken her engagement with would not be available later even if she found out later that he was the best, for by then he would be married to the second fairest maiden in the village.

The solution she reached, with the help of "Dear Abby" was to make a numerical rating of 0 to 100 to be given to each fellow after one month of engagement. If the rating of her fiance i were above a certain threshold θ_i get married to him, otherwise go on to the next fellow. If none of them suits her, she could still join the Marine Corps, though she is not as inclined to become a Woman Marine Officer as to settle down as a plain housewife. She thought that the last alternative was worth 20 points.

Unfortunately "Abby" has not told her how to determine her θ's, the threshold criteria to be used in judging the three fellows. What should they be to assure her the highest expected happiness in her future?

Example Problem 7.3. Industrial version I. The LJB Coffee Company has received a shipment of a ton of coffee beans smuggled in from Cuba. The company may inspect the quality of the beans and sell them as they are, or process them into ground coffee and sell them under their brand. If the quality as ground coffee were not satisfactory, it could be processed further into instant coffee. As a final resort, the product can always be made into caffein tablets "Never Doz" and always sold to local universities' graduate students for a total profit of $20.

Assume that each stage of process will cost $10 and that the price at which the product can be sold at each stage is directly proportional to its quality. Furthermore, the quality is distributed uniformly between 0 and 100 ($0 to $100) and is independent of the quality at other stages.

What would be the best decision criteria for processing this shipment and what would be its expected value?

Example Problem 7.3. Industrial version II (Ronald Howard's problem on carpenters, MIT, 1959). The personnel department of a company advertised for a position to be filled and four applicants applied for the position. The union requires that these must be interviewed in the order of arrival and each applicant must be hired or told to go elsewhere before the end of the interview when a new applicant is brought in. The company needs to have this position filled so that at least the last man will have to be hired. What will be the best criteria for the four applicants if the interview scores are uniformly distributed between 0 and 100?

Interpretation of Problem 7.3. All three versions are essentially three variations of a same problem. Moreover, the dynamic programming solution to this problem is very similar to the problem faced by a TLU in our nerve system. When a TLU keeps receiving the same signal several times within a short interval, we have the impression that it adjusts its own threshold to suits the exogenous inputs. For example, when we immerse our hand in a hot water bucket, the initial impulses of signals travelling down the nerve fibers are felt more strongly than the subsequent pulses which are transmitted after the initial impulses collapsed and recovered the nerve fibers to their excitable state (see Chapter I, p. 24).

Dynamic programming solution is akin to the way TLU's face their signals. We start from the last "empty state ϕ," of Mary been settled, coffee having been sold, position failled, and no signal for the neuron. We progress next to the state of Marine Corps, Never Doz, last applicant, and the first set of impulses arriving at the TLU. This process is repeated until either a steady-state condition has been reached or the number of states exhausted. At each stage, the threshold is readjusted regardless of whether more signals are coming in or not. Thus, we may look upon dynamic programming as an adaptive approach based on non-anticipation. This principle of optimality in terminology of a communication engineer, acts as a matched filter for a physically realizable system. Dynamic programming has the effect of creating an optimum weighting function that has the form of signal "running backward" starting from the fixed time t_1. A filter with this effect in communication engineering is called a matched filter. [Of course, this is but one case of a matched filter. A matched filter is a filter that maximizes signal-to-noise ratio. For a discussion of matched filters, readers are referred to Davenport and Root (1958, p. 244), Middleton (1960, p. 714).] More will be discussed on the topic of Information after Example 7.4.

Calculus solution of 7.3. The solution of this Problem 7.3 is orthodox and could make use of calculus. We shall take the most complete example, industrial version I, and solve for this particular problem. Others will be considered as its special cases.

At a state i, where i is the number of remaining states, we have the alternative b of accepting the particular product or alternative a of going on to another process. If the threshold were θ_i on the scale of 0 to 100 on which the quality of the product were to be distributed uniformly, then the probabilities will be given by:

$$p^a_{ji} = \frac{\theta_i}{100} \quad \text{and} \quad p^b_{ji} = 1 - p^a_{ji} = 1 - \frac{\theta_i}{100}$$

The quality, if accepted, will range uniformly between θ_i and 100:

$$r^b_{\phi i} = \theta_i + \tfrac{1}{2}(100 - \theta_i) = 50 + \tfrac{1}{2}\theta_i$$

If the product is rejected, the quality will be the expected quality of the product of the next process minus the quality (also the cost) lost by the processing, say c = \$10. Thus,

$$r^a_{ji} = -c = -10 \quad \text{and} \quad V_j, \quad \text{or} \quad R^k_{ji} = -c + V_j$$

and finally:

$$V_i = (\frac{\theta_i}{100})(V_j - c) + (1 - \frac{\theta_i}{100})(50 + \tfrac{1}{2}\theta_i)$$

Differentiating once with respect to θ_i and equating it to 0, we obtain:

$$0 = \frac{dVi}{d\theta_i} = \frac{1}{100}\,[- \theta_i + (V_j - c)]$$

or

$$\theta_i = V_j - c \qquad \text{where } c = \$10.$$

For this value of θ_i, we obtain the maximum for V_i

$$V_i = 50 + \frac{(Vj - c)^2}{200}$$

The solution is as shown on Figure 7.10, and means: accept the first process if its quality is better than 57.5, accept the second product if its quality is better than 50, and accept the third for 20 or better, otherwise push Never-Doz. The expected value of the shipment of coffee beans is 60.8 in this manner, instead of only 50.0 if the beans were to be sold without checking for possible future processing.

The dilemma of Mary is a same problem, except that $c = 0$. Its reduced tableau is shown on Figure 7.10 together with those of industrial versions. Marry the first fellow if he scores better than 63.52, marry the second for 52 or better, marry the last for 20 or above, otherwise join the Marine Corps. Since $c = 0$ (we assume that the enjoyment she receives during the courtship annuls any disadvantage), the expected value is a lot higher than in the coffee case, 70.17 instead of 60.8. The morale of the story is that a girl should date more, or better, marry the first person who solved this puzzle for her.

The last case of personnel department is similar. We have to hire the last fellow for an expected value of 50 if the first three do not qualify.

Dynamic programming solution of 7.3. Dynamic programming may bypass the need for calculus, that is if we are careful. It is very important in dynamic programming that all the management

1st Fellow	3	70. 17				
2nd Fellow	2	0 <63.52	63. 52			
3rd Fellow	1		0 ≤52	52. 00		
Marine Corps	0			0 <20	20. 00	
Future Decided	ϕ	81. 76 >63.52	76 >52	60 >20	20 ≥0	0. 00

Figure 7. 10. a. Mary's problem.

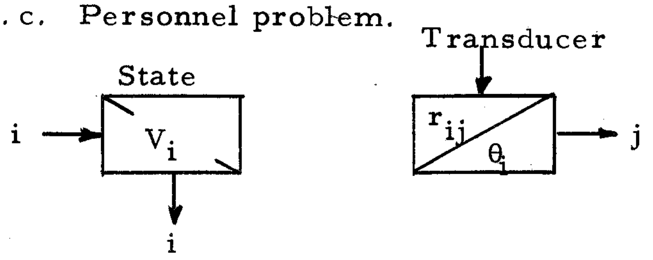
Coffee Beans	3	65. 53				
Coffee Grinds	2	-10 <51.17	61. 17			
Instant Coffee	1		-10 <40.5	50. 50		
Never Doz	0			-10 <10	20. 00	
Sold	ϕ	80. 6 >51.17	75. 3 >40. 5	55 > 10	20 ≥ 0	0. 00

Figure 7. 10. b. LJB coffee problem.

1st Applicant	4	74. 17				
2nd Applicant	3	0 <69.53	69. 53			
3rd Applicant	2		0 <62.50	62. 50		
4th Applicant	1			0 <50.00	50. 00	
Position Filled	ϕ	84. 76 >69.53	81. 25 >62.50	75 >50.00	50 ≥ 0. 0	0. 00

Figure 7. 10. c. Personnel problem.

Legend:



alternatives have been considered. For example, what were the alternatives facing the LJB Coffee Company management when it has received the shipment from Cuba? To accept to sell as is if its quality is above θ_3 and to process it into ground coffee if below θ_3? If we think so, then, we have been trapped. This is not really a management choice, it is actually Mr. Nature's turn to play. He can specify what the quality of coffee should be, but not us. The real choice that the management has is whether or not to let Mr. Nature have the pleasure of deciding the quality of beans or not. In other words, the management has the alternatives of whether or not subject the beans to the test and let Mr. Nature decide if they should fall below the threshold value or above the threshold value. The threshold value is essentially the rule of the game that we decide before the play starts. We cannot let the outcome of this game influence the rule to be used in the game. If we did, we will again be violating our non-anticipation law.

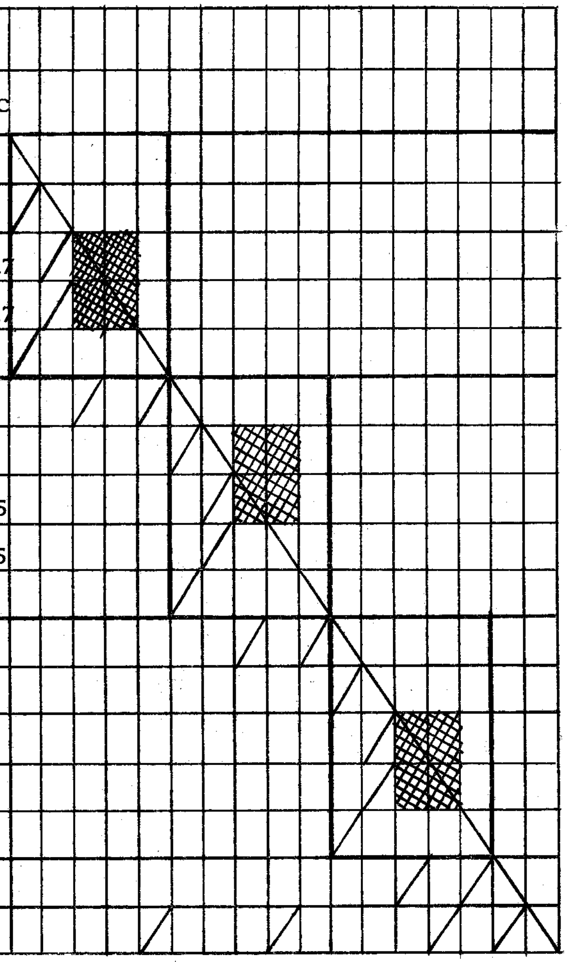
In the case of dynamic programming, θ's are determined by first computing the "Do Not Test" case: $V_i^{(2)} = V_j - c$. Then, in the case of testing, it is obviously advantageous to catch those that are above that expected value without inspection and to sell them. Thus, $\theta_i = V_j - c$. The detail of the work is shown on Figure 7.11.

State i	Alt k	Out j		P_{ji}	$r_{ji}^{(k)}$	v_j	$R_{ji}^{(k)}$ +		$V_i^{(k)}$ $\sum_j$	V_i $\max_k$	$\theta_i = V_j - c$
3			Coffee beans							65.3	
	1		Test	1					65.29	*	
		2	$< \theta_3$	0.5117	$-c = -10$	V_2 61.17	51.17	26.184			$<$ 51.17
		ϕ	$> \theta_3$	0.4883	50 + 1/2 61.17 80.575	V_ϕ 0	80.575	39.345			$>$ 51.17
	2	2	Do Not Test	1	-10	61.17	51.17	51.17	51.17		
2			Coffee Grinds							61.17	
	1		Test	1					61.17	*	
		1	$< \theta_2$	0.405	$-c = -10$	V_1 50.5	40.5	16.40			$<$ 40.5
		ϕ	$> \theta_2$	0.595	50 + 1/2 50.5 75.25	V_ϕ 0	75.25	44.77			$>$ 40.5
	2	1	Do Not Test	1	-10	50.5	40.5	40.5	40.5		
1			Instant Coffee							50.5	
	1		Test	1					50.5	*	
		0	$< \theta_1$	0.10	$-c = -10$	V_0 20	10	1			$<$ 10
		ϕ	$> \theta_1$	0.90	50+1/2 θ_1 55	V_ϕ 0	55	49.5			$>$ 10
	2	0	Do Not Test	1	-10	20	10	10	10		
0		ϕ	Never Doz	1	20	0	20	20	20	20	
ϕ			Sell	1	0	0	0	0	0	0	

0.5117 + 0.4883 = 1

0.405 + 0.595 = 1

0.10 + 0.90 = 1



$$\text{LJB Coffee Problem} \left\{ \begin{array}{l} r_{ji}^{1} = -c = -10 \\ V_0 = \ \ 20 \end{array} \right\} \text{Given}$$

Figure 7.11. LJB coffee problem tableau.

Utility Hypotheses

Price of freedom. The implications that can be brought on from the results of Problem 7.3 are far greater than just those of dynamic programming techniques. In order to appreciate what we have produced, let us plot the results on a graph as shown on Figure 7.12.

The shape of resulting curves, especially that of LJB Coffee problem, is a striking facsimile of what we have been calling a utility function.

The saturation point for the diminishing return is reached at:

$$V = 50 + \frac{(V - c)^2}{200}$$

or, completing the square and taking the negative root:

$$V = c - (200c) + 100 \qquad \text{for } 0 \leq c \leq 50$$

which comes out to be:

$$V = 100 \text{ for } c = 0$$

$$V = 65.28 \text{ for } c = 10$$

$$V = 50 \text{ for } c = 50$$

If c were greater than 50, the problem is not even worth considering (i.e. don't test coffee beans, sell them as they are).

Whether this is actually a utility function or not is a disputable but intuitively plausible hypothesis. The utility of Mary's charm would asymptotically reach 100, if more boys were available: i.e.

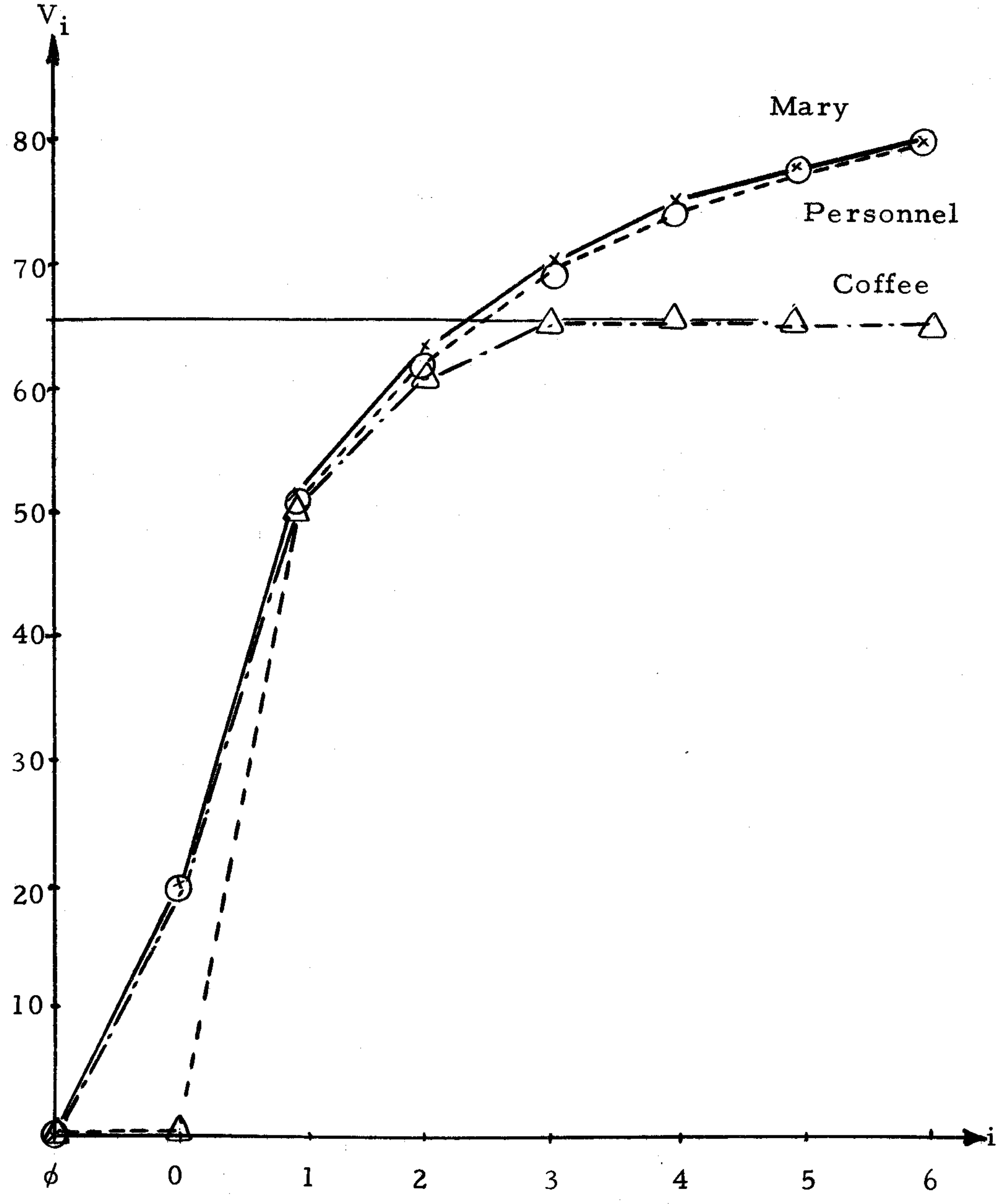


Figure 7.12. Threshold problems solution curves.

she could wait for ever until a mate with 100 rating appears and meanwhile enjoy her courtships fully.

In the case of LJB Coffee, regardless of the ingenuity of its engineering department to devise more ways of converting their product into profitable merchandise, the utility of having more means will not be greater than 65. 28 percent.

This limitation is similar to what we experience daily. We have a limitation as to how much we can enjoy ourselves at any particular time. At each moment, we have to decide which of the possible alternatives of actions we shall take at the next moment. If we have been working hard, we would like to take a rest. If we have been sleeping, perhaps we would like to get up instead.

Having more resources means that we will have more alternatives. But the utility of having too many is not really so welcomed. Most Americans are happy to be either a Democrat or a Republican, to watch either American League or National League, or tune into one of three major TV networks. It all stems from the fact that we can consider at most "7 ± 2" numbers of alternatives at once, and that we judge each alternative on a dichotomic basis. We create a set of elements of what we like, and for each element we decide whether it does or does not belong to the set. Each element is usually a feature of a particular alternative, and each alternative will be considered from several points of view. And yet, we are very uncertain to the actual effectiveness of the chosen alternative to satisfy our wants. This decision process is usually a task that requires some amount of mental concentration as well as some physical effort (examining various merchandises, measuring sizes, or just plain watching and listening to TV commercials). After we have examined the seventh decision or so, we are tempted to make the final choice from what we have already examined. Coincidentally, $i = 7$ would be a reasonable cut-off point of Figure 7.12.

> The general economic meaning of the term <u>utility</u> is the power to satisfy human wants. The utility that an object has for an individual is determined by him. Thus the utility of an object, like its value, is not inherent in the object itself but is inherent in the regard that a person has for it.
>
> Utility and value in the sense here used are closely related. The utility that an object has for a person is the satisfaction he derives from it. Value is an appraisal of utility in terms of media of exchange (Thuesen and Fabrycky, 1964, p. 18).

Larger the value of c is, faster the saturation is reached. In our example, Mary is a gay, charming, and joyful type that fully enjoys her boyfriends' company ($c = 0$). If she were a shy type to whom meeting a boy was more of a chore than pleasure, the value of c would have been positive.

The recursive expression can be made to look like:

$$Z_{j+1} = \alpha + \frac{1}{2} Z_j^2$$

by letting $Z_i = (V_i - c)/100$ and $\alpha = \dfrac{50 - c}{100}$ so that all evaluations will be made on a scale of zero to one, rather than zero to one hundred. The expansion of the expression in terms of j does present a series that may perhaps be forced into a more meaningful form. Bellman and Freyfus use :

$$r_i(x) = v_i [1-(1-e^{-a_i/x})^x]$$

for utility $r_i(x)$ or resource x, from an item i with market competition level of a_i and maximum potential profit v_i (Bellman and Dreyfus, 1962, p. 56).

More research is needed in this area before any more concrete remark can be made.

<u>Discount factors</u>. In order to proceed with the next example, we should mention one more possible approach to utility functions.

One factor that does take utility into account and which does seem to be working rather well, is the discount factor. Originally, the discount factor was computed on a discrete basis to compensate for the "time utility of money" at the beginning (or end) of each financial period.

$$\text{Discrete discount factor} = (1+r)^{-t}$$

where r = interest (or discount) rate, t = number of periods.

For a continuous case, it was found that a continuous factor could serve just as well:

$$\lim_{t \to \infty} (1+r)^{-t} = e^{-rt} = \text{Continuous discount factor.}$$

The actual difference between the two can be made obvious by expanding both expressions in McLaurin Series (or Taylor's):

$$f(r) = f(0) + rf'(0) + \frac{r^2}{2!}f''(0) + \frac{r^3}{3!}f'''(0) + \ldots$$

For continuous case:

$$e^{-rt} = 1 - rt + \frac{(rt)^2}{2!} - \frac{(rt)^3}{3!} + \frac{(rt)^4}{4!} - \ldots = \sum_{i=0}^{\infty} (-1)^i \frac{(rt)^i}{i!}$$

and in discrete case:

$$(1+r)^{-t} = 1 - rt + \frac{r^2[t(t+1)]}{2!} \quad \frac{r^3[t(t+1)(t+2)]}{3!} \quad \frac{r^4[t(t+1)(t+2)(t+3)]}{4!} + \ldots$$

$$= \sum_{i=0}^{\infty} (-1)^i \frac{r^i}{i!} \prod_{j=0}^{i} (t+j)$$

The actual differences for the cases of five percent and ten percent are shown in Figure 7.13.

The distinction between the discrete and continuous discount is reminescent of the difference between a Laplace Transform and a Z-Transform. If the discount factor were to be a time utility of money, perhaps the role of r and t should be perfectly interchangeable as in the case of the continuous discount factor.

Now, if our hypothesis that the coefficient rt (or -rt) indicates a time utility of money were indeed true, then we could extend our hypothesis to convert the resource utility of money in a similar manner. Thus, the utility of a fund $F = Ae^{-rt}$ could be characterized by its natural logarithm:

$$\ln(F) = \ln(Ae^{-rt})$$

rt	t	r	e^{-rt}	$(1+r)^{-t}$	$(1-r)^{t}$	t	r	e^{-rt}	$(1+r)^{-t}$	$(1-r)^{t}$
.05	1	5%	.95123	.95238	.95000					
.10	2	5%	.90484	.90703	.90258	1	10%	.90484	.90909	.90000
.50	10	5%	.60653	.61391	.59879	5	10%	.60653	.62092	.59049
1.00	20	5%	.36788	.37689	.35842	10	10%	.36788	.38554	.34868
2.00						20	10%	.13534	.14864	.12158

Figure 7.13. Discount factors comparison.

$$= \underset{\text{resource utility}}{\ln(A)} \quad + \quad \underset{\text{time utility}}{(-rt)}$$

This will lead to the concept of exchangeability between the time utility and the resource utility. Thus, two funds F_1 and F_2 can be equated, if:

$$\ln(A_1) + (-r_1t_1) = \ln(A_2) + (-r_2t_2)$$

or

$$\ln(A_1/A_2) = -(r_2t_2 - r_1t_1)$$

As an example, we let $r_1 = 5\%$, $t_1 = 10$, $r_2 = 10\%$, $t_2 = 10$: then, from Figure 7.13, we have: $A_2 = 0.60653\ A_1$ as it should be.

Noisy Communication Problem

In order to show the advantage of the logarithmic concept of utilities and to demonstrate how information theory keeps reappearing from behind the scene in systems theory, we shall now discuss the problem of "Efficient Gambler," originally presented by Bellman and Kalaba (1965, p. 86).

Example Problem 7.4.a. Perfect information. Suppose that an unscrupulous gambler operates by receiving over a Walkie-talkie, information concerning the outcomes of sporting events. If he has x dollars initially, and if his utility is a logarithmic function of x, he may bet his x dollars for double or nothing and expect to make a gross profit of 2x dollars or ln(2x) utils.

$$V_{max} = \ln(2x) = \ln(x) + \ln(2)$$

Example Problem 7.4.b. Noisy information. Suppose that the information received was disturbed with background noises and statics because of the limited capability of the Walkie-talkie. The gambler wishes to take his chance anyway but hesitates to bet his entire fortune (x). In order to safeguard his possible misfortune, he wishes to make a wager of y dollars only and use the difference (x-y) as a hedge. If his a priori probability of winning were p, how much should he bet?

Example Problem 7.4.c. Uncertain information. Suppose that the gambler has received a message (without noise) over the telephone about the present-state of the game a few minutes before the end of the game. According to the information, one team has a large lead over the other and the chance is slim (1-p) that the opponent will win. Using this information, how much should the gambler bet?

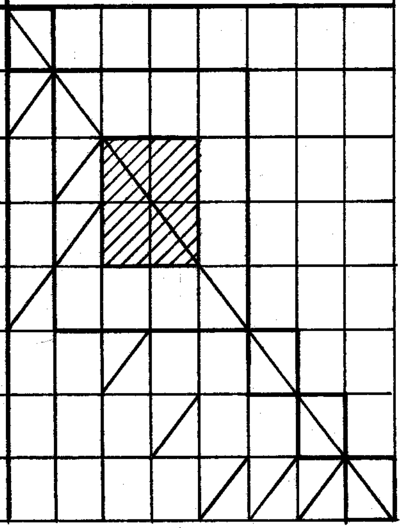
Solution to Problems 7.4.b and c. Obviously, as far as the gambler and his unsuspecting customers are concerned, it makes very little difference whether the case is b or c. The tableau is shown on Figure 7.14. We obtain in the case of gamble (if $p \geq \frac{1}{2}$):

$V_1^1 = p \ln(x+y) + (1-p) \ln(x-y) \equiv E$ (E is used for convenience)

The maximum is then obtained by:

$$0 = \frac{dE}{dy} = p \frac{d(x+y)}{(x+y)dy} + (1-p) \frac{d(x-y)}{(x-y)dy}$$

							$R_{ji}^{(k)}$		$V_i^{(k)}$	V_i
i	k	j		P_{ji}	$r_{ji}^{(k)}$	V_j	+	.	Σ	max
1			Before							
	1		Bet	1						If $p > \frac{1}{2}$
		2	Win	P	0	ln(x+y)	ln(x+y)	pln(x+y)	pln(x + y) + (1-p)(x-y)	ln(2x) + plnp + qlnq
		3	Lose	(1-p)	0	ln(x-y)	ln(x-y)	(1-p)ln(x-y)		
				(1-p) + p = 1						
	2	ϕ	No Bet	1	ln(x)	0	ln(x)	ln(x)	ln(x)	If $p < \frac{1}{2}$
2		ϕ	Won	1	ln(x+y)	0	ln(x+y)	ln(x+y)	ln(x+y)	ln(x+y)
3		ϕ	Lost	1	ln(x-y)	0	ln(x-y)	ln(x-y)	ln(x-y)	ln(x-y)
ϕ			After	1	0	0	0	0	0	0



Efficient Gambler

Figure 7.14. Efficient gambler - single decision.

$$0 = \frac{p}{x+y} - \frac{1-p}{x-y} \text{ or } y = (2p-1)\,x \qquad \begin{array}{l}\text{maximum if } p > \frac{1}{2} \\ \text{minimum if } p < \frac{1}{2}\end{array}$$

The maximum value of V_1^1 is then:

$$E_{max} = V_{1max}^1 = p \ln[x + (2p-1)x] + (1-p) \ln[x - (21-1)x]$$

$$= p \ln(2px) + (1-p) \ln(2x-2px)$$

$$= \ln(x) + \ln(2) + p \ln(p) + (1-p) \ln(1-p)$$

We recognize the first two terms from the case 7. 4.a of perfect information. The last two terms are both negative since $\frac{1}{2} \leq p \leq 1$. If we had used the logarithm of base 2 instead of e, we could have recognized p $\log_2(s)$ as the equivocation due to the lossiness of the channel and $(1-p) \log_2(1-p)$ as the irrelevance due to the noisiness of the channel.

The transformation between the Naperian[19] (natural) logarithm of base e = 2. 718281828459045 . . . and the logarithm of base two is merely a matter of adjusting for convention: i. e. a simple multiplication by ln2. $\ln(x) = \ln(2)\log_2 x$ where $\ln(2) = 0.30103$.

<u>Information utility.</u> The surprising results which stemmed out of our simple hypothesis of a logarithmic utility function does not stop here. If we rewrite E_{max} in terms of the conversion factor to adjust to bits, we obtain:

---

[19] Named after the Scotch mathematician John Napier, the inventor of Napier's Bones and Lord of Merchiston (1550-1617).

$$E_{max} = \ln(x) + \ln(2) + p\,\ln(2)\log_2(p) + (1-p)\ln(2)\log_2(1-p)$$

$$= \ln(x) + \ln(2)[1 + p\log_2(p) + (1-p)\log_2(1-p)]$$

$$= \binom{\text{resource}}{\text{utility}} + \binom{\text{potential utility gained from}}{\text{additional information}}$$

Just as our utility hypothesis has allowed us to separate the time utility from the resource utility, we have here an expression that clearly indicates the incremental utility the gambler has obtained thanks to the private communication from the "clairvoyance," in this case, his friends at the race tracks.

Since utility is an expectation of use to be gained from the future applications, it is always a "potential" resource. If utility is to be a "potential," then where is its corresponding "kinetic" form? Obviously, that must be the "value" that we have used to find the "utility" form. "Dollars" derive their values from their circulations. Owning a dollar bill has the potential utility for the circulation value of a dollar. If we can identify "circulation" as an expression of "freedom", then we have come back to the reasoning that led us to these exciting results.

<u>Van Neumann's foresight</u>. If we consider p as the individual probability for gaining satisfaction, then the resource utility part of the function may be considered as a universal function. The individuality of utility function is due to the other factors, such as: the way a person values his time, whether he is optimistic or pessimistic

(larger or smaller p), what he can do with the money (larger the number of desirable alternatives, larger the value of p will be), and so on. The resource utility, on the other hand, will be determined by the society, and may be considered invariant within the same society at a given time. (Obviously, government, for example, could change the utility of the resource within the country. However, we shall assume this to be a gradual, time-consuming change.)

The apparent connection between the "utility" and "entropy" is particularly significant. Von Neumann[20] and Morgenstern, yet unaware of the subsequent development of the information theory based on the concept of "entropy," had obviously guessed intuitively the close similarity between the heat-temperature relationship and utility-information relationship. We quote from their book, originally written in 1943:

> ... It seems however that even a few remarks may be helpful, because the question of the measurability of utilities is similar in character to corresponding questions in the physical sciences.
>
> Historically, utility was first conceived as quantitatively measurable, i.e. as a number. Valid objections can be and have been made against this view in its original, naive form. It is clear that every measurement--or rather every claim of measurability--must ultimately

---

[20]Von Neumann has been quoted as the one responsible for applying the work of Boltzmann and Gibbs in statistical mechanics to the study of information content of physical measurements with particular reference to quantum theory, thus creating (with Szilard) the foundation for Shannon's Information Theory (1925-1930) (MIT, 1959).

be based on some immediate sensation, which possibly cannot and certainly need not be analyzed any further.[21] In the case of utility the immediate sensation of preference--of one object or aggregate of objects as against another--provides this basis. But this permits us only to say when for one person one utility is greater than another. It is not in itself a basis for numerical comparison of utilities for one person nor of any comparison between different persons. Since there is no intuitively significant way to add two utilities for the same person, the assumption that utilities are of non-numerical character even seems plausible. The modern method of indifference curve analysis is a mathematical procedure to describe this situation.

All this is strongly reminescent of the conditions existant at the beginning of the theory of heat: that too was based on the intuitively clear concept of one body feeling warmer than another, yet there was no immediate way to express significantly by how much, or how many times, or in what sense.

This comparison with heat also shows how little one can forecast a priori what the ultimate shape of such a theory will be. The above crude indications do not disclose at all what, as we now know, subsequently happened. It turned out that heat permits quantitative description not by one number but by two: the quantity of heat and temperature. The former is rather directly numerical because it turned out to be additive and also in an unexpected way connected with mechanical energy which was numerical anyhow. The latter is numerical, but in a much more subtle way; it is not additive in any immediate sense, but a rigid numerical scale for it emerged from the study of the concrodant behavior of ideal gases, and the role of absolute temperature in connection with the entropy theorem.

The historical development of the theory of heat indicates that one must be extremely careful in making negative assertions about any concept with the claim

---

[21] Such as the sensations of light, heat, muscular effort, etc., in the corresponding branches of physics.

> to finality. Even if utilities look very unnumerical today, the history of the experience in the theory of heat may repeat itself, and nobody can foretell what ramifications and variations.[22] And it should certainly not discourage theoretical explanations of the formal possibilities of a numerical utility (Von Neumann and Morgenstern, 1964, p. 16).

Transform Theory

How can we put our utility hypotheses to work? The basic concept we have introduced in the form of "logarithmic" expression may be restated as: "The change of utility is a function of the relative importance of the change of the value," or in the case of time-resource trade-off: $d(rt) = \frac{dA}{A}$ which becomes: $rt = \ln A$. By any chance, do we already possess such transformation pairs? Fortunately, the answer seems positive. Economists may argue that what we are doing is reinventing funds-flow formulae, and indeed the results become equivalent in the case of time-resource relationships, but we believe that our outlook will provide insights not available or not as obvious in the classical theory. And it has the side benefit of introducing both Laplace and Z-transforms needed to show further applications of tableaux.

---

[22] A good example of the wide variety of formal possibilities is given by the entirely different development of the theory of light, colors, and wave lengths. All these notions too became numerical, but in an entirely different way (Von Neumann and Morgenstern, 1964).

Definition 7.1. Dirac[23] delta function. The Dirac delta function $\delta(t-\tau)$ is defined by:

$$\int_{-\infty}^{\infty} \delta(t-\tau)f(t)dt = f(\tau)$$

which causes a sampling of f(t) at the instant $t = \tau$.

Definition 7.2. Unit-impulse train: A unit-impulse train δ_T is defined as an ideal sampler which becomes effective with intervals T:

$$\delta_T(t) = \sum_{-\infty}^{\infty} \delta(t-nT)$$

Definition 7.3. Laplace[24] transform. Laplace transformation of a function f(t) which is zero for $t < 0$ is defined by:

$$F(s) = \int_{0-}^{\infty} f(t)e^{-st}dt \;\; = L[f(t)]$$

F(s) exists if f(t) is sectionally continuous and if it does not grow faster than exponentially. In other words, for $t \to \infty$, there must

---

[23] Named after Paul Adrien Maurice Dirac, a French Quantum Physicist who introduced the concept in 1947 in connection with the orthonormality of eigenfunctions (Schiff, 1955; Mandl, 1957; etc.).

[24] Named after P. S. Laplace, a French mathematician. His major work "Théorie analytique des probabilités" was first published in 1812.

exist a finite number c such that $f(t)\ e^{-ct}$ is bounded.[25]

Now, if we take the Laplace transform of the Dirac delta function, then we would obtain:

$$L[\ \delta(t-T)] = \int_0^{\infty} \delta(t-T)e^{-st}dt = e^{-sT}$$

Similarly for a unit-impulse train, we have:

$$L[\ \delta_T(t)] = \int_0^{\infty} \sum_{n=0}^{\infty}(t-nT)e^{-st}dt = \sum_{n=0}^{\infty} e^{-nTs}$$

$$= 1 + e^{-Ts} + e^{-2Ts} + \ldots = (1 - e^{-Ts})^{-1}$$

By replacing sT by rt, we recognize that we have a series of time values with a discount rate e^{-rt}.

At each period nT from the time t = 0, the discount rate e^{-rnt} will allow us to convert a single future payment A_n to the present worth $A_n e^{-rnT}$. This rate is sometimes called "single-payment present-worth factor" for continuous compounding interest (Thuesen and Fabrycky, 1964, p. 83).

---

[25] Of course, Laplace transform is but a special case of integral transformation

$$F(s) = \oint_C K(s, t)f(t)dt$$

with Kernel $K(s, t) = e^{-st}$. This is essentially a mapping process from the t-(time) domain to s-(frequency) domain. A Kernel is a subgroup of the time domain which consists of all elements mapped into the identity element in the s-domain; in this case, just one element e^{-st}. C defines the path of integration.

The Laplace transform of a unit-impulse train can then act as a "carrier for amplitude modulation"[26] or a discrete "time-value series" that will convert the set of future payments (or incomes) to the present-worth.

In general, a function f(t) can be sampled by this ideal sampler to yield: f*(t) as shown in figure 7.15:

$$f^*(t) = f(t)\delta_T(t) = f(t)\sum_{n=0}^{\infty}\delta(t-nT) = \sum_{n=0}^{\infty} f[nT\,\delta(t-nT)]$$

Its Laplace transform becomes:

$$F^*(s) = L[f^*(t)] = L\left[\sum_{n=0}^{\infty} f(nT)\delta(t-nT)\right]$$

and using the results obtained for the Laplace transform of Dirac delta function, we can write this as:

$$F^*(s) = \sum_{n=0}^{\infty} f(nT)e^{-nTs}$$

which bears a striking ressemblance with the defining equation:

$$F(s) = \int_0^{\infty} f(t)e^{-st}dt$$

Example Problem 7.5. Rent-a-car system. An airport rent-a-car system estimated that the yearly income from the rent of a car

---

[26] Again, we come face to face with the information theory of Shannon: PCM (pulse code modulation) and PPM (pulse phase modulation) (Shannon, 1949, p. 3).

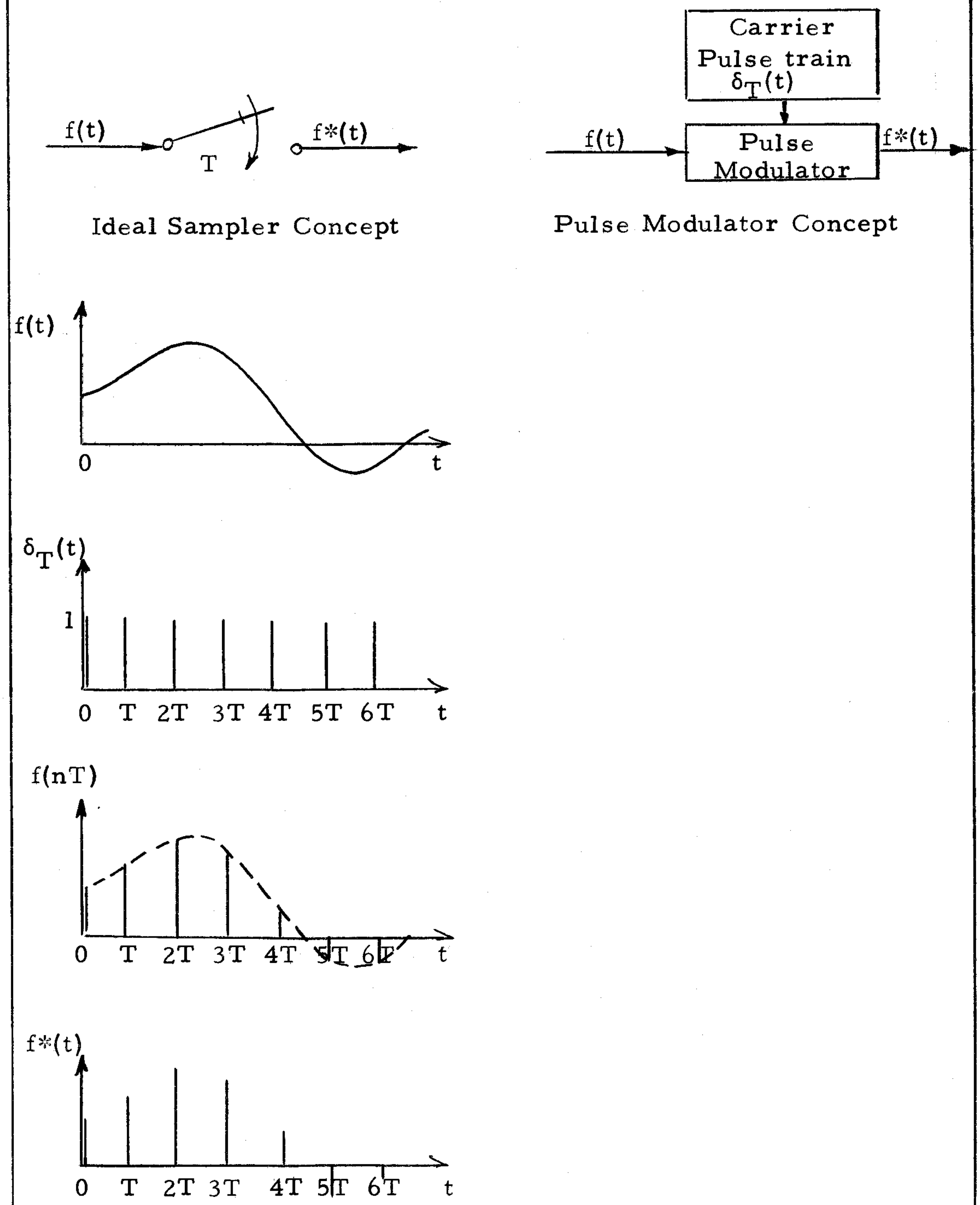


Figure 7. 15. Sampling concepts.

is inversely proportional to its age. If the discount factor were ten percent and its first year income were \$10,000 what would be the present worth of a new car?

Solution to Problem 7.5. It is interesting to note that our sampler is incapable of recognizing whether the original function is f(5) or f(nT). In both cases, we obtain the same $F^*(s)$. In this particular example, we may use: $f(nT) = A_n$ where $A_n = 10,000/(n-1)$. Thus, setting $s = 0.10$, we obtain:

$$F^*(s=10\%) = \sum_{n=0}^{\infty} A_n e^{-nTs} = A_o + A_1 e^{-.1} + A_2 e^{-.2} + A_3 e^{-.3} + \dots$$

$$= 10,000 + 4,524 + 2,046 + 926 + \dots$$

What we have done in fact is to turn Laplace transform into a generating function. But we already know a generating function of a very similar nature, namely the Z-transform.

Definition 7.4. Z-transform. The Z-transform of a variable f(n) which takes on values at n = 0, 1, 2, 3, . . . is defined as[27]

$$F(z) = \sum_{n=0}^{\infty} f(n) z^{-n}$$

The striking thing is that Z-transform is obtained from $F^*(s)$ by setting: $z = e^{Ts}$ or $s = \ln(s)/T$

[27]Some authors use z^{+n} instead of z^{-n}. Very unfortunately Ronald Howard happens to be one of them (Howard, 1960, p. 7).

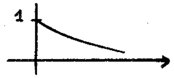
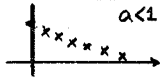
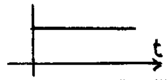
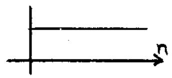
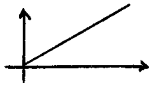
	LAPLACE					Kuo, Freeman z	Howard $\mathbf{Z}$
	$F(s)$	$f(t)$	$f(t)$	$f(n)$	$f(n)$	$F(z)$	$F(z^{-1}=Z)$
Addition	$F_1(s) \pm F_2(s)$	$f_1(t) \pm f_2(t)$			$f_1(n) \pm f_2(n)$	$F_1(z) \pm F_2(z)$	$F_1(Z) \pm F_2(Z)$
Scalar Factor	$aF(s)$	$af(t)$			$af(n)$	$aF(z)$	$aF(Z)$
Integration	$\frac{1}{s} F(s)$	$\int_o^t f(t)dt$			$f(n-1)$	$z^{-1}F(z)$	$ZF(Z)$
Differentiation	$sF(s) - f(o)$	$\frac{d}{dt} f(t)$			$f(n+1)$	$z\ F(z) - f(0)$	$Z^{-1}\ F(Z) - f(0)$
Multiple Integration	$\frac{1}{s^k} F(s)$	$\int \cdots \int_k f(t)dt_1\,dt_2 \cdots dt_2$			$f(n-k)$	$Z^{-k}F(Z)$	$Z^{k}F(Z)$
Exponential Decay	$\frac{1}{s+a}$	e^{-at}			a^n	$\frac{z}{z-a}$	$\frac{1}{1-aZ}$
Unit Impulse	1	$\delta(t)$			$\delta(0)$	1	1
Unit Step 1	$\frac{1}{s}$	1			1	$\frac{z}{z-1}$	$\frac{1}{1-Z}$
Unit Impulse Train	$\frac{1}{1-e^{-Ts}}$	$\sum_{n=0}^{\infty} (t-nT)$			$\sum_{n=0}^{\infty} (t-nT)$	$\frac{z}{z-1}$	$\frac{1}{1-Z}$
Unit Ramp t or n	$\frac{1}{s^2}$	$1t$			$1n$	$\frac{z}{(z-1)^2}$	$\frac{Z}{(1-Z)^2}$
	$\frac{1}{(s+a)^2}$	te^{-at}			na^n	$\frac{aZ}{(z-a)^2}$	$\frac{aZ}{(1-aZ)^2}$
	$F(s+a)$	$e^{-at}f(t)$			$a^n f(n)$	$F(aZ)$	$F(aZ)$

Figure 7.16. Transforms.

Ts was found to be the rt in the discount factor which we have already identified as "Time Utility," while ln(z) is what we have hypothesized as the "Resource Utility."

To convert utility into value, or vice versa, transforms may be compared using the specific time function (figure 7.16).

Econometrics

Definition 7.5. Econometrics. "Econometrics is the application of a specific method in the general field of economic science in an effort to achieve numerical results and to verify economic theorems" (Schumpeter, 1933).

Before leaving the subject of utility hypotheses, let us briefly examine the field of econometrics to see whether there is any evidence that supports our utility hypotheses.

We have postulated that there can be a cardinal function of utility which is composed of resource utility, time utility, and information utility. Each component is evaluated as a logarithmic function and is additive, while the values derived from the utility are exponential and multiplicative.

Henry Schultz's study of demand function for wheat. In 1938, Henry Schultz published a book entitled "The Theory and Measurement of Demand" in which he has succeeded to derive the statistical demand function for wheat in the United States for the period

1921-1934 using Bureau of Labor statistics (Tinter, 1952, p. 38).

The least square multiple regression analysis yielded the following result:

$$\ln(x) = 1.0802 - 0.2143 \ln(p) - 0.00358\, t - 0.00163\, t^2$$

x is the quantity of utilized wheat less seed in bushels per capita. p is the deflated (using Bureau of Labor statistics) farm price of wheat in cents per bushel. t is the time with 1928 taken as t = 0.

The above equation can be rewritten as:

$$5.05 = 4.66 \ln(x) + \ln(p) + 0.0167\, t + 0.0076\, t^2$$

$$5.05 = \ln(p{*}x^{4.66}) + (1.67\%)t + \ln(2)\, 0.00252\, t^2$$

Is it assuming too much to interpret this relationship as:

Utility = resource utility + time utility + information utility?

Obviously, we should await the results of much more extensive study before speculating the validity of our hypothesis. However, in the event that our hypotheses do become verified, we can expect a very interesting study of relationships between the information utility and such factors as: the economic prospect of a nation, optimistic or pessimistic outlook of an industry, raising or lowering of interest rate by the Federal Reserve Board, and so on.

Other econometric studies. Unfortunately, this treatise is not a proper place to extend our study of utility hypotheses any further. Let us merely mention that other studies conducted by F. V. Waugh (1935) and R. Frisch (1932) seem to reveal similar logarithmic characteristics. Undoubtedly, there must be other statistics which may or may not support our hypotheses.

Other utility function studies. The question of integribility of the total demand in a model in which each customer acts according to a cardinal utility function and has a fixed monetary income, has been studied by E. Eisenberg of Hughes Aircraft Company (Veinott, 1965, p. 296). The axiomatization of utility in the line of von Neumann and Morgenstern was pursued by the Stanford Value Theory project by Patricks Suppes, Muriel Winet and others (Veinott, 1965, p. 284).

Ergodic Processes

Efficient Gambler Recursion.

If the efficient gambler of Problem 7.4 were to continue his gambling every day, the principle of optimality will yield a recurrence relationship:

$$V_n = \underset{0 \leq y \leq x}{\text{maximum}} \, [p \, V_{n-1}(x+y) + (1-p)V_{n-1}(x-y)]$$

with V_1 as calculated on page 252 ($V_1 = V^1_{1_{max}} = E_{max}$).

We can rewrite this as:

$$V_1(x) = \ln(x) + K$$

with

$$\begin{cases} K = \ln(2) + p \ln(p) + (1-p)\ln(1-p) & \text{if } p \geq \tfrac{1}{2} \text{ (gamble)} \\ k = 0 & \text{if } p \leq \tfrac{1}{2} \text{ (do not gamble)} \end{cases}$$

Then, Bellman shows inductively that (Bellman and Kalaba, 1965, p. 88)

$$V_n(x) = \ln(x) + nK$$

by writing (assuming the above relationship to hold for n):

$$\begin{aligned} V_{n+1}(x) &= \underset{0 \leq y \leq x}{\text{maximum}} \, [p \ln(x+y) + nK] + (1-p) [\ln(x-y) + nK] \\ &= \ln(x) + (n+1)K \end{aligned}$$

Since the relationship is true for $n = 1$, the assertion has been proven.

The name "efficient gambler" now becomes clear. We can safely venture to assume that Bellman and Kalaba mean an "asymptotically efficient" gambler.

Markov Chains

Defintion 7. 6. Markov chain. A system with r states is said to be a Markov chain, iff the conditional probability of transition from the present state i to the next state j does not depend on how the system arrived at state i.

Definition 7. 7. Ergodic process. If all states in a Markov chain communicate and if a state i exists such that $p_{ii} \geq 0$, then the Markov chain is said to be ergodic.

From our discussion in Chapter IV (Theorem 4. 2) we note that a tableau must be irreducible to be ergodic. The additional requirement is that there is a trapping state with self-feedback loop $p_{ii} > 0$.

Definition 7.8. Ergodic set. A subset of states in a system such that once the system entered a state in the subset can never be left is called an ergodic set. Its states are referred to as recurrent states.

Thus, if a tableau contains a subtableau meeting the requirement of Definition 7.7, that portion forms an ergodic set.

If a Markov system is a true description of a system, it must contain at least one ergodic set. In other words, regardless of what state the system is in, that state must be included in the Markov system.

Theorem 7.2. Limiting states. For an ergodic system, all the limiting state probabilities are independent of the system's initial states.

Howard's policy improvement scheme. Ronald Howard used this property to formulate:

$$V(n \text{ large}) = ng + V$$

which became the basis for his value iteration (Value Determination Operation) and policy improvement scheme (Policy Improvement Routine) developed in his Sc. D. thesis of 1958 (Howard, 1960). Unfortunately, we have no time to discuss his theory in conjunction with the tableau using z-transform. Instead, we will show Howard's toymaker's problem and equipment maintenance problem solutions on tableaux in Figure 7.17 and 7.18. The procedure followed is identical to that used in previous examples, except that iteration is used for $n = 0$ (the last state), $n = 1$ (one before last), and so on. We shall state the problems and briefly comment on the solutions. Both examples, as well as the one for the efficient gambler, are obviously ergodic processes asymptotically converging toward Howard's formula.

Example Problem 7.6. Toymaker's problem. A toymaker wishes to sell his business within three months. Each month, he can be in either of two states: successful or unsuccessful. If he is successful, he can advertise or not advertize during the following one month. If he is unsuccessful, he can either spend money in R + D or let the fate decide how successful he will be the following month. The transition probabilities and expected profits (or expenditures) are as shown on the tableau. What policy should he adopt to present a most prosperous picture of his business at the time of sales?

Solution to Problem 7.6. For the first two months of the three, he should advertise if he is successful, and he should do R + D if unsuccessful. The last month, do neither.

Example Problem 7.7. Maintenance problem. A taxi-cab company expects to trade in their fleet for new models within six months. The cabs may be either running in optimal condition

i State	k Altern	j Outcome	Control Engr's Notation / Guthrie's Notation	F(x, t) $P_{ij}^{(k)}$	J(k) $r_{ij}^{(k)}$	n = 0 $V(n)_j$	+	·	J Σ	U^* $J(i)^*$ max	n = 1 I $v(n)_j$	+	·	J Σ	u^* J^* max	n = 2 I v_j	+	·	J Σ	u^* J^* max
1			Successful							k=1 6					k=2 8.2					k=2 20.22
	1		No Adv						6	U^*				7.5					9.3	
		1	Success	$\frac{1}{2}$	9	0	9.	4.5			6	15	7.5			8.2	17.2	8.6		
		2	Unsuccess	$\frac{1}{2}$	3	0	3.	1.5			-3	0	0			-1.7	1.3	6.7		
	2		Advertize						4					8.2	U^*				20.22	U^*
2			Success	$\frac{4}{5}$	4	0	3.	1.2			6	10	8			8.2	12.2	9.8		
	1		Unsuccess	$\frac{1}{5}$	4	0	4	0.8			-3	1	.20			-1.7	2.3	4.6		
2			Unsuccessful							k=1 -3					k=2 -1.7					k=2 +0.23
	1		No R + D						-3	U^*				-9					-7.4	
		1	Success	$\frac{2}{5}$	3	0	3	1.2			6	9	1.8			8.2	11.2	44.8		
		2	Unsuccess	$\frac{3}{5}$	-7	0	-7	-4.2			-3	-10	-6			-1.7	8.7	-53.2		
	2		R+D						-5					-1.7	U^*				0.23	U^*
		1	Success	$\frac{7}{10}$	1	0	1	.7			6	7	4.9			8.2	9.2	6.44		
		2	Unsuccess	$\frac{3}{10}$	-19	0	-19	-5.7			-3	-22	-6.6			-1.7	-2.1	-6.21		

Figures shown for n = 0 only

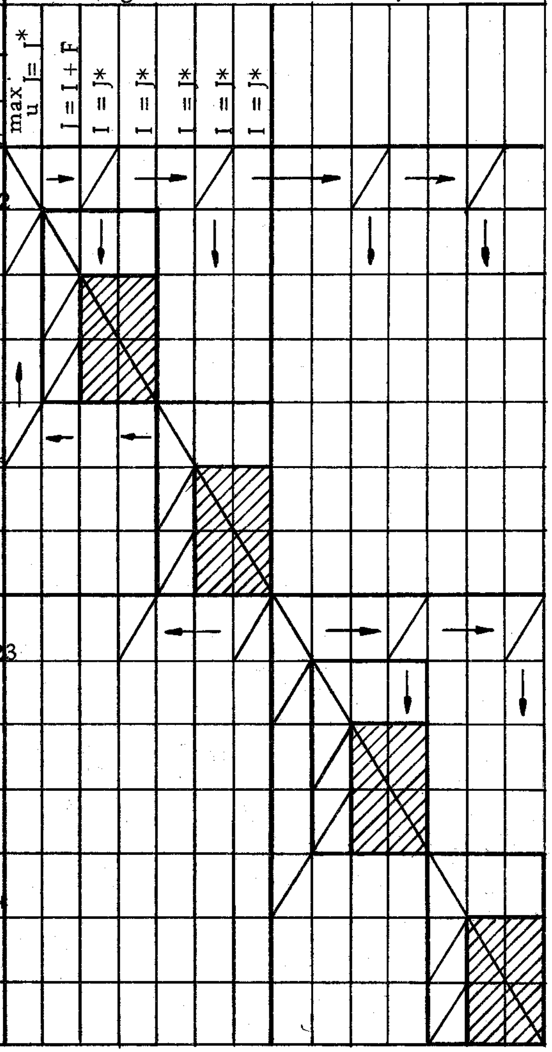


Figure 7.17. Toy-maker's tableau

(economic gasoline consumption, etc.) or in suboptimal conditions. In either case, the monthly service may be obtained from a quality garage for a premium fee, or from a cheap neighborhood garage for nominal fee. The fees and the expense of running under the given condition for a month is shown on the Tableau in Figure 7.18. From the data given on the transition probabilities determine the optimal policy for the last six months.

Solution to Problem 7.7. Use quality service for the first two months regardless of the condition. Use quality service on cabs with good running condition only during the next three months. Use the cheapest service in both cases in the last maintenance job.

The value of g is found to be 11.67.

Control Theory

Convolution Summation Method vs. State Method

There are two ways in which discrete time control problems are formulated. The first is a weighted sequence of past inputs to determine the present state of system. The second is the familiar state-representation. It can be shown that both methods are equally easily represented by tableaux. In addition, we note that we may include the z-transform concept and use z^{-1} as a delayor relating state n to n + 1. The problem of the controllability and observability of a

i	k	j				n = 0 V_j	n = 0 +	n = 0 ·	n = 0 Σ	n = 0 min	n = 1 V_j	n = 1 +	n = 1 ·	n = 1 Σ	n = 1 min	n = 2 V_j	n = 2 +	n = 2 ·	n = 2 Σ	n = 2 min	n = 3 V_j	n = 3 +	n = 3 ·	n = 3 Σ	n = 3 min
1			Optimum Performance							6					16					27					38.42
	1		U_1 Cheap Maint						6	U_1^*				18					30					4.2	
		1	Optimum	$\frac{1}{2}$	6	0	6	3			6	12	6			16	22	11.0			27.0	33.0	17.0		
		2	Sub-optimum	$\frac{1}{2}$	6	0	6	3			18	24	12			32	38	19.0			44.7	50.7	25		
	2		U_2 Quality Maint						7					16	U_2^*				27	U_2^*				38.4	U_2^*
		1	Optimum	$\frac{3}{4}$	7	0	7	5.25			6	13	9.8			16	23	17.2			27.0	34.0	25.5		
		2	Sub-optimum	$\frac{1}{4}$	7	0	7	1.8			18	25	6.25			32	39	9.8			44.3	51.7	12.1		
2			Sub-optimum Performance							18					32					44.67					56.78
	1		U_1 Cheap Main						18	U_1^*	*			32	U_1^*				44.7	U_1^*				56.5	U_1^*
		1	Optimum	$\frac{1}{3}$	18	0	18	6			6	24	8			16	34	11.3			27.0	45.0	15.0		
		2	Sub-optimum	$\frac{2}{3}$	18	0	18	12			18	36	24			32	50	33.3			44.7	62.7	41.8		
	2		U_2 Quality Main						21					33					45					56.8	
		1	Optimum	$\frac{1}{2}$	21	0	21	10.5			6	27	13.5			16	37	18.5			27.0	48.0	24.0		
		2	Sub-optimum	$\frac{1}{2}$	21	0	21	10.5			18	39	19.5			32	53	26.5			44.7	65.7	32.8		
			Difference					12					16					17.67					18.36		

Figure 7.18. Taxi-cab maintenance problem. (Continued on p. 272).

i	k	j				n = 4 V_j	+	•	Σ	min	n = 5 V_j	+	•	Σ	min
1			Optimum Performance							50.02					61.68
	1		U_1 Cheap maint						54.4					65.31	
		1	Optimum	$\frac{1}{2}$	6	38.4	44.4	22.2			50.02	56.02	28.01		
		2	Sub-optimum	$\frac{1}{2}$	6	56.8	62.7	31.4			68.60	74.60	37.30		
	2		U_2 Quality Maint						50.02	U_2^*				61.68	U_2^*
		1	Optimum	$\frac{3}{4}$	7	38.4	46.3	34.6			50.02	57.02	42.80		
		2	Sub-optimum	$\frac{1}{4}$	7	56.8	63.5	15.9			68.60	75.60	18.98		
2			Sub-optimum Performance							68.60					80.31
	1		U_1 Cheap Maint						68.65					80.40	
		1	Optimum	$\frac{1}{3}$	18	38.4	56.4	18.8			50.02	68.02	22.67		
		2	Sub-optimum	$\frac{2}{3}$	18	56.8	74.7	49.8			68.60	86.60	57.73		
	2		U_2 Quality Maint						68.60	U_2^*				80.31	U_2^*
		1	Optimum	$\frac{1}{2}$	21	38.4	59.4	29.7			50.00	71.02	35.51		
		2	Sub-optimum	$\frac{1}{2}$	21	56.8	79.8	38.9			68.60	89.60	44.8		
									18.85					18.63	

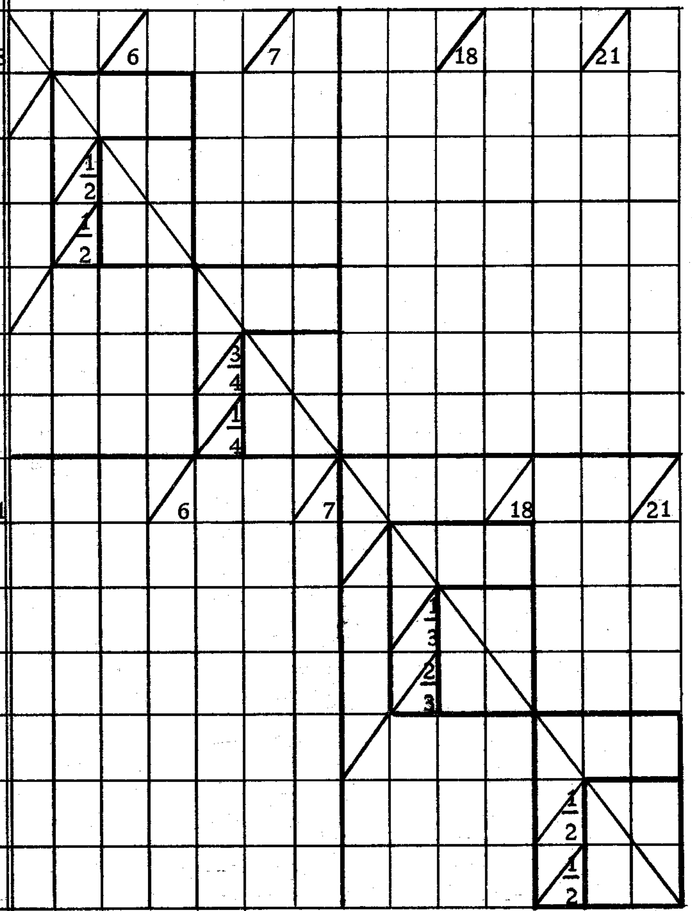


Figure 7.18. Taxi-cab maintenance problem. (Continued from p. 271).

system then becomes the simple problem of finding the eigenvalues and diagonalizing the tableau. These are shown on Figures 7.19 and 7.20.

Continuous Cases

In general a control problem can be classified as follows:

1. static vs. dynamic.
2. continuous vs. discrete.
3. deterministic, stochastic, adaptive.

A problem is considered continuous if the time interval $t_o \leq t \leq t_1$ is examined continuously.

The following classification of problems and techniques are due to D. Michael Intrilligator of the Economic Department of the University of California at Los Angeles.

The formulation was based on his presentation at the "Modern Systems Theory and Applications to Large Scale Systems Seminar" held at U. C. L. A. during the Summer 1966.

Control Problem

Find $\underline{u}(t)$ that will maximize J, the objective functional of $\underline{u}(t)$:

$$\underset{\underline{u}(t) \epsilon U}{\text{maximize}}\; J[u(t)] = \int_{t_0}^{t_1} I(\underline{x}, \underline{u}, t)dt + F[\underline{x}(t_1), t_1]$$

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}, t)$$

$$\underline{x}(t_o) = \underline{x}_o$$

	$u(k)$			$x_3(k)$		$x_2(k)$		$x_1(k)$	$y(k)$	
$u(k)$										
	1									
$x_3(k+1)$		$\frac{4}{3}$		$\frac{26}{24}$		$-\frac{9}{24}$		$\frac{1}{24}$		
$x_3(k)$			z^{-1}							
$x_2(k+1)$		3		1						
$x_2(k)$					z^{-1}					
$x_1(k+1)$		7				1				
$x_1(k)$							z^{-1}			
				24		-17		3		
$y(k)$									1	

$$x_1(k+1) = x_2(k) + 7u(k)$$

$$x_2(k+1) = x_3(k) + 3u(k)$$

$$x_3(k+1) = \frac{1}{24} x_1(k) - \frac{9}{24} x_2(k) + \frac{26}{24} x_3(k) + \frac{4}{3} u(k)$$

$$y(k) = 3x_1(k) - 17x_2(k) + 24x_3(k)$$

$$\begin{bmatrix} 26/24-\lambda & -9/24 & 1/24 \\ 1 & -\lambda & 0 \\ 0 & 1 & -\lambda \end{bmatrix} = 0$$

$\lambda_1 = 1/2$; $\lambda_2 = 1/3$; $\lambda_3 = 1/4$

Figure 7.19. Original control tableau.

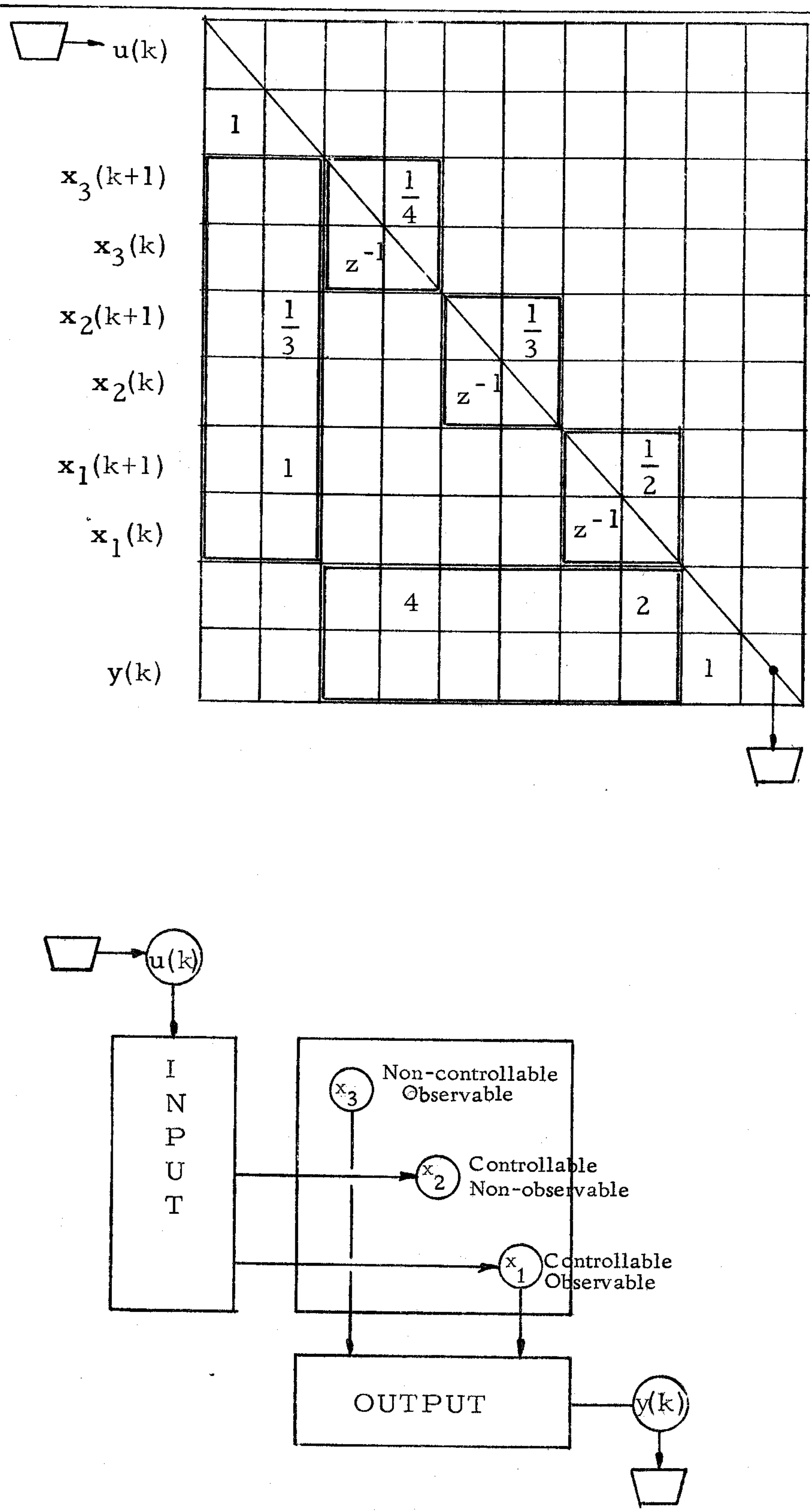


Figure 7. 20. Diagonalized tableau.

This was first presented by). Bolza in 1909 in his "Vorlesungen Über Variationsrechnung" ("Lecture on calculus of variation," Tebner, Leipzig) and is called problem of Bolza. When $F \equiv 0$, and $J = \int I(\underline{x}, \underline{u}, t)dt$, this is called problem of Lagrange, while when $I \equiv 0$, and $J = F[\underline{x}(t_1), t_1]$ is called problem of Mayor.(Cf. figure 7.21),

Special Cases of the Functional J

Static case. $t_o = t_1$ (e.g. linear programming problems).

$$J(\underline{u}) = F(\underline{x}, t) = F(\underline{x})$$

Minimum time. (e.g. critical path scheduling, PERT).

$$J = -(t_1 - t_o) \qquad \text{or minimize } (t_1 - t_o)$$

This can arise either:

by setting $I = -1$ and $F = 0$ or,

by setting $I = 0$ and $F = -t_1$ (assume $t_o = 0$).

Maximize average value.

$$J = \frac{1}{t_1 - t_o} \int_{t_0}^{t_1} I(\underline{x}, \underline{u})dt$$

by setting $F = 0$ and $I = \frac{1}{t_1 - t_o} I(\underline{x}, \underline{u})$

Maximum present value. (e.g. engineering economics).

$$J = \int_{t_0}^{t_1} e^{-rt} I(\underline{x}, \underline{u})dt$$

where e^{-rt} is the discounted factor. r is the interest rate.

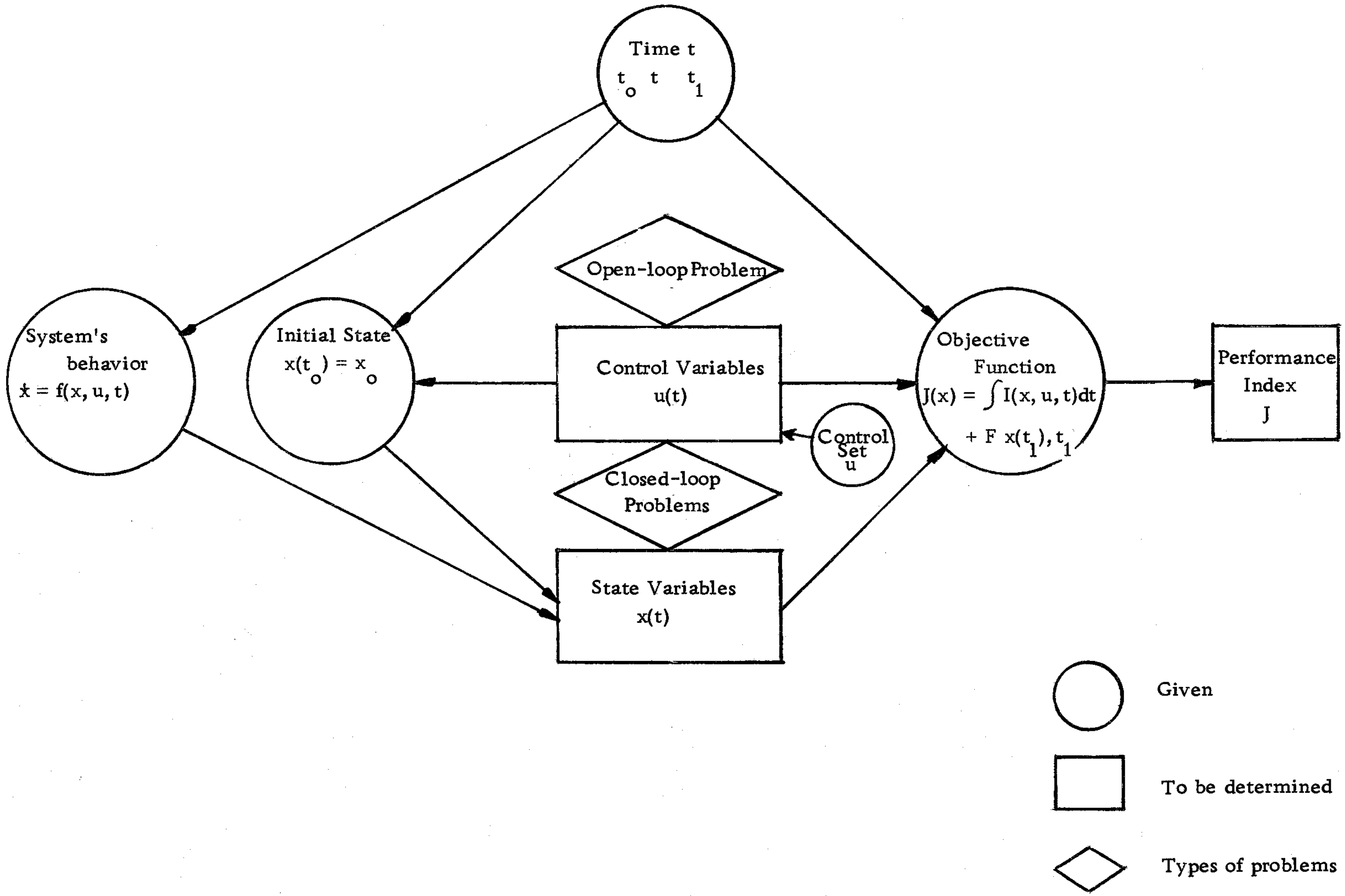


Figure 7.21. Intrilligator's Diagram.

Servomechanism problem.

$$\bar{x}(t) = \text{desired state at time } t.$$

$$J = -\int_{t_0}^{t_1} \psi[\underline{\bar{x}}(t) - \underline{x}(t)]\,dt$$

where $z = [\underline{\bar{x}}(t) - \underline{x}(t)]$ is the error at time t.

(i) $n = 1$ $\psi(\underline{z}) = \underline{z}^2$ minimum square error.

(ii) $n > 1$ $\psi(\underline{z}) = \underline{z}'W\underline{z}$ where $\underline{z}'W\underline{z}$ is the weighted square.

Methods of Solution ($t_0 \neq t_1$)

Calculus of variations (classical approach). Calculus of variations deals with the optimization of problems of Lagrange; namely, the cases where $F \equiv 0$ and $J(\underline{u}) = \int_{t_0}^{t_1} I(\underline{x}, \underline{u}, t)dt$, where u(t) is continuous and $\dot{\underline{x}} = \underline{u}$. The standard solution is obtained from the Euler's equation:

$$\frac{\partial I}{\partial \underline{x}} = \frac{d}{dt}\left(\frac{\partial I}{\partial \dot{\underline{x}}}\right)$$

Dynamic programming (attributed to Bellman, 1957). This is based on the familiar principle of optimality. The objective functional becomes:

$$J^*(\underline{x}, t) = \underset{\underline{u}(t)\epsilon U}{\text{maximum}} \left[I(\underline{x}, \underline{u}, t)dt + J^*(\underline{x}+d\underline{x}, t+dt) \right]$$

where J^* must be continuous.

Expanding the latter part of the integral in Taylor's series, we obtain:

$$J^*(\underline{x}+d\underline{x}, t+dt) = J^*(\underline{x}, t) + \frac{\partial J^*}{\partial \underline{x}} d\underline{x} + \frac{\partial J^*}{\partial t} dt + \ldots$$

This can be used to write the original functional as:

$$J^*(\underline{x}, t) = \underset{\underline{u}(t) \epsilon U}{\text{maximum}} \left[I(\underline{x}, u, t) + J^*(\underline{x}, t) + \frac{\partial J^*}{\partial \underline{x}} \; \frac{d\underline{x}}{dt} \quad \frac{\partial J^*}{\partial t} \right]$$

Since $J^*(\underline{x}, t)$ is a term added to each factor evaluated for maximum, we can cancel it with the term on the left. By writing $\frac{d\underline{x}}{dt}$ as $\underline{f}$, we obtain Bellman's equation:

$$-\frac{\partial J^*}{\partial t} = \underset{\underline{u} \; \epsilon \; U}{\text{maximum}} \left[I(\underline{x}, \underline{u}, t) + \frac{\partial J^*}{\partial \underline{x}} \cdot \underline{f} \right]$$

Of course, the problem is to find the solution to the Bellman's equation using presently available methods and equipments, a dilemma dubbed "curse of dimensionality" by Bellman.

Maximum principle. This method was developed in the U. S. by M. R. Hestenes while at U. C. L. A. in 1958, and by L. S. Pontryagin in Russia in 1957 and which appeared in the article, "Some mathematical problems arising in connection with the theory of optimal systems of automatic control, " in the Proceedings of the Academy of Science, USSR (vol. 11, pp. 107-117, 1957).

The control problem is solved by introducing the dual variables, called variously auxiliary variables, Lagrange multipliers, costates, shadow prices, etc.

$$p(t) = [p_1(t), \; \ldots, \; p_n(t)]$$

Thus, the problem of maximizing J has become an alternate problem of maximizing a Hamiltonian (so named because of its obvious similarity to the classical mechanics' Hamiltonian, see for example, Goldstein, 1950, p. 217, p. 243, etc.) by the proper choice of $\underline{u}^*(t)$, the optimal control vector.

$$H(\underline{x}, \underline{u}, \underline{p}, t) = I(\underline{x}, \underline{u}, t) + \underline{p} \cdot \underline{f}(\underline{x}, \underline{u}, t)$$

$$\underset{u \in U}{\text{maximum}}\, H(\underline{x}, \underline{u}, \underline{p}, t) = H^*(\underline{x}, \underline{p}, t) = H(\underline{x}, \underline{u}^*, \underline{p}, t)$$

The procedure is similar to the method of Lagrange multiplier we have used to verify the Maximum Information Theorem.

Since $f(x, u, t) = \dot{x}$, we can find p from the following 2n equations:

$$\dot{\underline{x}} = -\frac{\partial H^*}{\partial \underline{p}} \quad \text{and} \quad \dot{\underline{p}} = -\frac{\partial H^*}{\partial \underline{x}}$$

with the given boundary conditions: $\underline{x}(t_o) = \underline{x}_o$.

The similarity between this approach and Bellman's becomes obvious at once:

$$\underset{\text{(Bellman)}}{-\frac{\partial J^*}{\partial t}} = \underset{\text{(maximum)}}{\max(H)} = \underset{u \in U}{\text{maximum}}\left[I(x, u, t) + \frac{\partial J^*}{\partial x} f\right]$$

An interesting discussion showing the relationships between the Pontryagin maximum principle and calculus of variations and dynamic programming is presented by Richard E. Kopp (Leitmann, 1962, p. 255).

VIII. PAST, PRESENT, AND FUTURE OF TABLEAU METHODS

'Beauty is truth, truth beauty,'
---this is all
Ye know on earth,
and ye need to know.
(From "Ode on Grecian Urn" by John Keats)

Past

Tableau Économique

F. Quesnay (1694-1774). The term "tableau" was first used by François Quesnay in 1758 to describe a mathematical model of a system. His original tableau dealt with the interdependence of economic activities in a farm and showed the repercussion that a given increment in output would have on the rest of wealth-producing activities. Later he published an expanded version of his tableau to show the entire French economy of his time. The reprint of his first book (published in Versailles, 1758) was later published by H. Higgs in London (1894) (Tintner, 1952, p. 63; Miernyk, 1965, p. 5).

> The current introduction of linear programming in economics appears to be an anachronism; it would seem logical that it should have begun around 1758 when economists first began to describe economic systems in mathematical terms. Indeed a crude example of a linear programming model can be found in the tableau economique of Quesnay, who attempted to interrelate the roles of the landlord, the peasant, and the artisan (Dantzig, 1963, p. 16).

L. Warlas. The concept of the general equilibrium of exchange is usually attributed to the 1874 work by Léon Walras "Éléments d'économie politique pure" in which an attempt was made to simultaneously determine all prices in an economy, including factors of production ("fixed technological coefficients") as well as the prices of finished goods (Miernyk, 1965). The concept of partial static equilibrium initiated by Quesnay was made into a general equilibrium theory by Walras, and the work was pursued by others such as Gustav Cassel of Sweden, Vilfredo Pareto of Italy, and G. C. Means of the United States. Means used simple regression techniques to derive relationships between economic variables such as production, employment, and construction.

W. W. Leontief. The modern Input-Output analysis based on tableaux economique is attributed to Professor Wassily W. Leontief of Harvard who developed a general theory of production along with the first complete input-output tableau for the American economy.

> Leontief's basic ideas were first published in his article "Quantitative Input-Output Relations in the Economic System of the United States," The Review of Economics and Statistics, XVIII (August 1936), 105-125. These ideas were expanded in other journal articles, and in 1941 Leontief's first book on input-output economics was published under the title The Structure of American Economy, 1919-1929. An expanded version of this book, covering the period 1919-1939, was published by Oxford University Press in 1951 (Miernyk, 1965, p. 6).

Figure 8.1 shows Leontief's tableau économique for the year 1939. The figures are in billions of dollars, and the tableau is essentially that laid out by Leontief, except, of course, for the eigen line. The exogenous inputs are listed to the left of the tableau, and the endogenous outputs, to the bottom. They are the total sums produced or consumed by the respective industries. "n. e. c." stands for chemicals, lumber, and wood industry, furniture, paper, printing, and construction. "Other industries" include banking, insurance, advertising, various services, rents, laundry, amusements, etc. The grouping and figures are from Tintner's book (1965, p. 64).

Matrix Theory

Linear models. The input-output analysis is but an application of matrix theory taking advantage of the linearity of the economic exchange model.

> While it is certainly dangerous to construct linear models of real-world problems, it is difficult to forego the esthetic pleasure involved in obtaining explicit solutions. How to reconcile the two is one of the challenges of model-building (Bellman and Dreyfus, 1962, p. 297).

It is probably not an overstatement to say that the entire linear theory is based on the Fundamental Theorem usually called "Fredholm's Alternatives."

Distribution of Outlays	Inputs	1	2	3	4	5	6	7	8	9	10	11
1. Agriculture and food	17.0					0.6		0.6	0.6		0.7	14.5
2. Minerals	3.8	0.1		1.2				0.2	1.3		0.9	0.1
3. Metal fabricating	12.3	0.7	0.1		0.3	0.1	0.3	1.1	2.1	0.3	4.2	3.0
4. Fuel and power	8.9	0.4	0.3	0.4		0.1	0.3	0.7	0.4	0.2	2.6	3.5
5. Textile, leather and rubber	7.0	0.1		0.3				0.2	0.1		0.8	5.4
6. Railroad transportation	4.3	1.3	0.3	0.4	1.0				0.5	0.1		0.7
7. Foreign trade	2.8	1.0	0.4		0.1	0.2			0.5		0.6	
8. Industries n. e. c.	19.2	0.9	0.1	0.4	1.0	0.5	0.6	0.4		4.6	5.2	5.6
9. Government (taxes)	13.8	1.1		0.2	0.2				0.1		9.7	2.6
10. Other industries	60.9	8.2	1.5	3.4	3.1	3.1	0.7	0.1	9.1	2.8		28.9
11. Household (consumption)	68.8	4.2	1.1	6.7	3.6	2.9	2.5		5.5	7.9	34.5	
Outputs	218.8	18.0	3.8	13.0	9.3	7.5	4.4	3.3	20.2	15.9	59.2	64.3

Figure 8.1. Leontief's tableau for 1939 U. S. economy.

Theorem 8.1. Fredholm's alternatives. For the system of equations:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= y_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= y_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= y_n \end{aligned}$$

or $AX = Y$, $(\sum_{j=1}^{n} a_{ij}x_j = y_j)$ with given coefficients a_{ij}'s, the following alternatives hold:

(1) Either it has one and only one solution X for each arbitrarily given vector Y, in particular the solution $X = 0$ for $Y = 0$; or

(2) Alternatively, the homogeneous equations arising from $AX = Y$ for $Y = 0$ have a positive number k of nontrivial (not identically zero) linearly independent solutions $X_1, X_2, \dots, X_k$, which may be assumed to be normalized. In the latter case, the transposed (dual) homogeneous system of equations:

$$A'X' = 0, \ (\sum_{j=1}^{n} a'_{ij}\, x'_j = \sum_{j=1}^{n} a_{ij}x'_j = 0) \qquad \begin{matrix}(i = 1, \dots, n) \\ (j = 1, \dots, n)\end{matrix}$$

where $a'_{ij} = a_{ji}$, also has exactly k linearly independent nontrivial solutions for just those vectors $X'_1, X'_2, \dots, X'_k$. The inhomogeneous system $AX = Y$ then possesses solutions for just those vectors Y which are orthogonal to $X'_1, X'_2, \dots, X'_k$. These solutions are determined only to within an arbitrary solution of the homogeneous system of equations, i. e. if X is a solution of the inhomogeneous system $AX = Y$ and X_m is any solution of the homogeneous system $AX = 0$, then $X + X_m$ is also a solution of the inhomogeneous system $AX = Y$ (adapted from Courant and Hilbert, 1953, p. 6).

Example Problem 8. 1. Exchange model. Assume a simplified economic tableau with three industries: food, clothing, and shelter. By dividing each exchange by the total output of the corresponding industry, we can obtain a technological coefficient tableau as shown in Figure 8. 2. We shall call this Tableau A. Then, the input vector X will yield $AX = Y$ output. From our discussion from Chapter I (page 38), we have:

$$\frac{Y}{X} = \frac{\text{Output value}}{\text{Input value}} = \lambda \geq 1$$

However, when the output and input are taken at exactly the same instant (zero time interval), or equivalently, when the system is static (independent of time), the ratio must approach unity. Thus, $X = Y$ for $\lambda = 1$. Therefore, $AX = X$, or $(A-I)X = 0$, and from the Fredholm's alternative (2), the determinant of $(A-I)$ must vanish. This is equivalent of an eigenvalue problem where $\lambda = 1$ in $(A-\lambda I)$.

No. 1	Food	0. 2	0. 3	0. 0
No. 2	Clothing	0. 0	0. 0	0. 4
No. 3	Shelter	0. 8	0. 7	0. 6

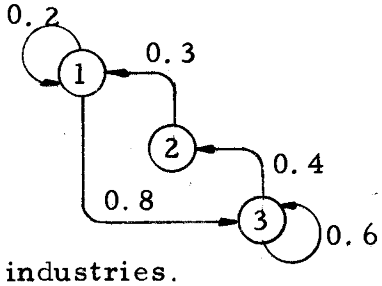


Figure 8. 2. Exchange model for three industries.

The eigenvector X turns out to be:

$$X = \begin{pmatrix} 0.15a \\ 0.4a \\ a \end{pmatrix} \qquad \text{with } a = X_m \text{ any positive number.}$$

Gale (1960) shows how this concept can be extended to tableau containing several irreducible subsets. Figure 8.3 shows how he re-arranges a tableau to produce irreducible subsets (very much like the Schmidt's Orthogonalization Process) and by finding eigenvectors for each subset, obtains the overall tableau vector.

	1	2	3	4	5	6	7
1	0.2	0.1		0.3			
2		0.2					0.5
3		0.2	0.5		0.9		
4						0.4	
5			0.5		0.1		0.2
6	0.8			0.7		0.6	
7		0.5					0.3

	1	4	6	3	5	2	7
1	0.2	0.3	0.0			0.1	0.0
4	0.0	0.0	0.4			0.0	0.0
6	0.8	0.7	0.6			0.0	0.0
3				0.5	0.9	0.2	0.0
5				0.5	0.1	0.0	0.2
2						0.2	0.5
7						0.5	0.3

$$\text{Eigen-vector } X = \begin{pmatrix} 0.15a \\ 0 \\ 1.8b \\ 0.4a \\ b \\ a \\ 0 \end{pmatrix}$$

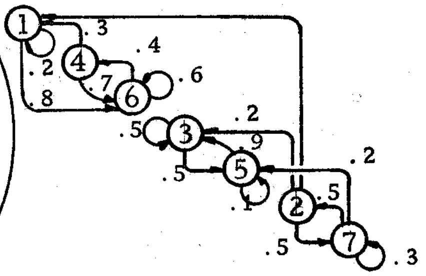


Figure 8.3. Gale's example (Gale, 1960, p. 271).

Linear Programming

Transportation tableau. The stepping-stone method used in transportation tableaux is very similar to the staircase chaining method we use in tracing the flow within a tableau. The transportation method is usually credited to T. C. Koopmans (Dantzig, 1963, p. 18), but A. Charnes, W. W. Cooper and many others are also involved in the development (Hadley, 1962) so that it is impossible to single out a person as an originator. The merit of the transportation tableau is obviously in the explicit use of Primal-Dual relationship in reaching the optimal solution.

Simplex tableau. On the other hand, very few people will dispute the origin of Simplex tableau. This is clearly the work of George Dantzig. One way of interpreting the Simplex method is as a very clever use of linearity in solving Lagrange Multiplier problems. Because of linearity, taking a derivative becomes simple division. Comparing the ratios will determine which of the $g_i = 0$ is the most constraining restriction. By using the Dual, we can decide which is the steepest gradient to the extremum.

Of course, there are many interpretations of Simplex method. We shall show, later in this chapter, how a Simplex problem could be solved within the framework of our tableau.

Present

Critical Path Scheduling

C.P.S. Tableau. The C.P.S. Tableau method was developed in 1964 to assist in the solutions of various problems connected with project scheduling. The paper presented at the AIIE national convention in 1966 (Riggs and Inoue, 1966) shows the close connection between CPS and dynamic programming, and tableau and Ford Fulkerson's network problems. In particular, CPS tableau was shown to be capable of generating its own cut network (dual network of the original problem) so that Ford-Fulkerson's min-cut max-flow algorithm could be applied. Until then, the only available method was a graphical technique by which a dual graph was lifted from the primal. This method had not only the disadvantage of being inaccessible to the operation by computer, but also proved to be impossible when the primal network was not planar (Vajda, 1961, p. 55; Inoue, 1964).

> Although the theoretical problem is thus solved, there is still the practical question of finding the minimal cut-set. Ford and Fulkerson have given a computational algorithm for finding the maximal flow. . . If the network is planar (in the two terminal sense) we can draw its dual, assign edge weights equal to those of the corresponding edges in the original, and use Moore's technique for finding the shortest path through a maze. Since the shortest path corresponds to a minimal cut-set, the problem is solved (Seshu and Reed, 1961, p. 273).

The algorithm we have derived works equally well for planar as non-planar networks. In essence, it is a method by which all communications between states are found (either by taking Neumann Inverse, or by using CPS tableau's chaining method) and non-communicated states (activities in CPS) which form the edges of the original network are identified as sets of states which may co-exist (corresquisite activities). This is accomplished by taking the "negative" image of the communicated tableau (either the result of Neumann Inverse or of chaining process). Thus, we will have an entry (1 or /) in the cell corresponding to an empty transducer in the communicated tableau, and no entry in the cells corresponding to existing transducers in the communicated tableau. If the chain method does yield a path in the new dual tableau, this path forms a part of the dual network.

The computer program for finding all dual paths for a given primal network (tableau size 10x10) is shown in the listing of Figure 8.4. Though the program is limited and is rather inefficient (FORTRAN), it is operational and has been used successfully on IBM 1620.

Before we leave the topic of computer, let us show another application of CPS tableau to computer programming:

Parallel processing.

> New methods, together with extensions and generalizations of old methods, are required to answer the new and more exacting questions of modern science and technology and, in particular, to make fuller use of that scientific factotum, the Sorcerer's Apprentice, the digital computer. This remarkable device, even in its infancy, with scant understanding on our part of its use and potential, has already significantly and indeed irrevocably altered the ground rules of mathematics and science (Bellman and Kalaba, 1965, p. 3).

```
C THIS PROGRAM WILL PRINT OUT THE TABLEAU SO THAT A DUAL NETWORK MAY BE
C FOUND ACCORDING TO THE ALGORITHM DEVELOPED IN THE DISSERTATION.
      DIMENSION TITLE(20), ACT(10, 10), JD(10), KD(10, 10)
C INITIALIZATION.
    1 DO 100 I=1, 10
      DO 100 J=1, 10
  100 KD (I, J)=0
      READ 10, ( TITLE(I), I=1, 20)
   10 FORMAT(20A4)
      PRINT 11, (TITLE(I), I=1, 20)
   11 FORMAT (20A4//)
      PUNCH 11, (TITLE(I), I=1, 20)
      PRINT 12
   12 FORMAT (15HINITIAL TABLEAU/)
      MAX=0
  200 READ 20, ID, CAP, (ACT(ID, I), I=1, 10), (JD(J), J=1. 10)
   20 FORMAT (12, F8. 2, 10A4, 1013)
      IF (ID)201, 202, 203
  201 PRINT 13, (ACT(I), I=1, 10)
      GO TO 200
  203 MAX=MAX+1
      DO 300 I=1, 10
      IF (JD(I))301, 300, 302
  301 KD(JD(I), ID)=1
      GO TO 300
  302 KD(ID, JD(I))=1
  300 GO TO 200
      PRINT 15, MAX, MAX
      PUNCH 15, MAX, MAX
   15 FORMAT (12HTABLEAU SIZE, 12, 1HX, 12//23H      ACTIVITY DESCRIPTION, 20X.
     131HCAPACITY ID 1 2 3 4 5 6 7 8 9 10)
      DO 400 IN=1, MAX
      PRINT 16, (ACT( IN, J=1, 10), CAP, ID, (KD(ID, I)I=1, MAX)
      PUNCH 16, (ACT(IN, J=1, 10), CAP, ID, (KD( ID, I)I=1, MAX)
   16 FORMAT(10A4. F8. 2, 13, 1012)
  400 CONTINUE
      IF (IEXIT) 410, 600, 410
  410 PRINT 1M
   17 FORMAT (///21HCORREQUISITES TABLEAU//)
      PUNCH 17
      DO 500 IL=1, MAX
      NI=MAX-IL
      DO 500 JL=1, MAX
      NJ=MAX-JL
      IF(K( NI, NJ ))501, 500, 501
  501 DO 500 KL=1, 10
      K(NI, KL)=K(NJ, KL)+K(NI, KL)
  500 CONTINUE
      IEXIT=1
      DO 530 I=1, 10
```

(continued on next page)

```
    DO 530 J=1, 10
    IF(K(I,J))530, 520, 530
520 K(I,J)=K(J,I)
530 CONTINUE
    GO TO 202
600 CONTINUE
    PAUSE
    GO TO 1
    END
```

Figure 8.4. FORTRAN II listing of dual network generation algorithm.

In electronic data processing where speed of execution of a program is one of the primary considerations along with the precision and reliability of its operation, it is being recognized that the computers of tomorrow (and some of today) will be of multi-processor construction. In essence, this means that there will be several processing units (each equivalent of a central processor (CPU) of today's computer) working together on a "pool" of works. The pool will contain all works to be done which are ready for the processors. Each processor will pick up a work and return the result to the pool. A control unit will use the result in producing further work available for processors, or send it to an output unit or a memory storage unit.

All in all, this resembles a cooperative job shop with several workers whose activities are coordinated by one foreman. There are several advantages to this scheme. The foreman may assign a same work to two individuals so that it may compare the results to make sure of their accuracy. When a worker is sick or absent for a periodic check-up, the factory will only suffer in speed of production and not in its capability.

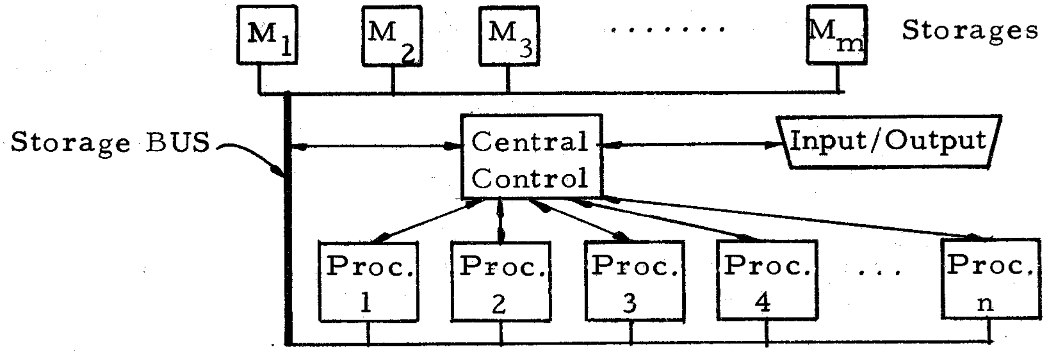


Figure 8.5. Machine organization for parallel execution (Hellerman, 1966: modified).

Parallel programming. When a computer has parallel processing capability, a series program cannot make a full use of this advantage. Traditionally, a statement written in a high-level programming language, such as FORTRAN, had to be changed into a chain of single-step machine operations.

With the advent of a parallel-processor computer, there will be a need for generating parallel programs that may be executed by several processors at once. H. Hellerman from IBM Corporation Systems Research Institute has proposed a "tableau" and "tree" approach to the algorithm of generating processor assignment of machine operation steps.

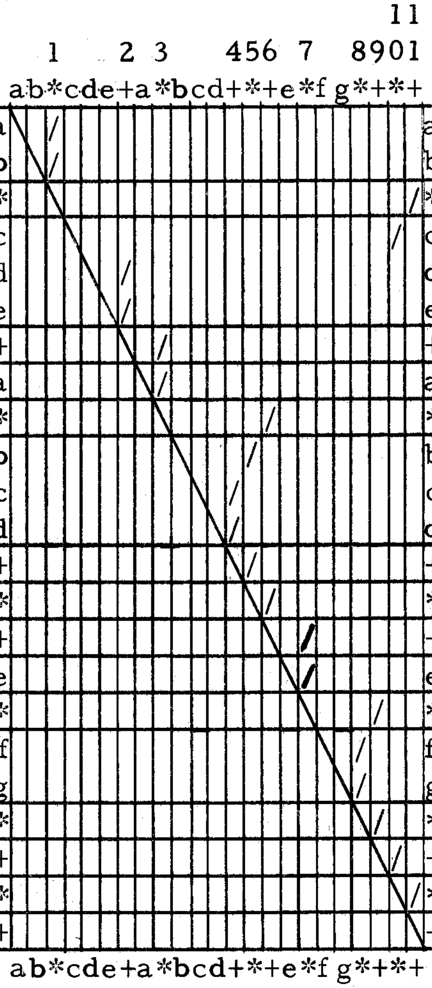
For example: $ab + \{[(d+e)a + (d+c)b]e + gf\}c$, his "tableau" will result in Figure 8.6.

a	b	*	c	d	e	+	a	*	b	c	d	+	*	+	e	*	f	g	*	+	*	+
		1*				2*		3*				4*	5*	6*		7*			8*	9*	10*	11*
/	/	1		/	/	1				/	/	1					/	/	1			
						/	/	2	/			/	2									
								/					/	3								
														/	/	4						
																/		/	5			
			/																/	6		
		/																		/	7	

The check marks (/) indicate the operands for the operation to their right numbered according to their level. The starred numbers are operation identification numbers. All operations in one level can be performed simultaneously. All operations corresponding to the smaller level numbers are supposed to have been performed previously.

Figure 8.6. Hellerman's tableau.

The concept of our Polish tableau can be applied directly to produce results obtained from Hellerman's tableau. This is shown in Figure 8.7 where ES, EF, LS, and LF are computed according to the algorithm developed for the CPS tableau (Inoue, 1964).



ES	ET	EF			LF	ET	LS
0	0	0	a	a	5	0	5
0	0	0	b	b	5	0	5
(0	1	1	1*	*1	6	1	5)
0	0	0	c	c	5	0	5
0	0	0	d	d	0	0	0
0	0	0	e	e	0	0	0
(0	1	1	2+	+2	1	1	0)
0	0	0	a	a	1	0	1
(1	1	2	3*	*3	2	1	1)
0	0	0	b	b	1	0	1
0	0	0	c	c	0	0	0
0	0	0	d	d	0	0	0
(0	1	1	4+	+4	1	1	0)
(1	1	2	5*	*5	2	1	1)
(2	1	3	6+	+6	3	1	2)
0	0	0	e	e	3	0	3
(3	1	4	7*	*7	4	1	3)
0	0	0	f	f	3	0	3
0	0	0	g	g	3	0	3
(0	1	1	8*	*8	4	1	3)
(4	1	5	9+	+9	5	1	4)
(5	1	6	10*	*10	6	1	5)
(6	1	7	11+	+11	7	1	6)

ES+ET=EF LF-ET=LS

ET: Instruction time = 0
Execution time = 1
in this example.

Figure 8.7. Polish tableau for parallel programming.

By letting ET (estimated time) for each operation to be one machine cycle, we have arrived at the same result as indicated by Hellerman.

Hellerman's tableau has levels corresponding to the EF (earliest finish) in the tableau above, and are referred to as "earliest-stage-assignment rule." Hellerman recognizes that sometimes it is preferrable to follow the "latest-stage-assignment rule" though he does not show how this is obtained. The levels corresponding to the "latest-stage" can be obtained directly from the Polish tableau by consulting the LF (latest finish) column. The two schedules were constructed by Hellerman as "execution trees" and are shown on Figures 8.8 .a and b.

Several advantages of the Polish tableau over Hellerman's tableau are obvious. First, it allows the scheduling of various machine operations of uneven lengths. It is hardly likely that a simple addition takes as much time as a multiplication or division. Second, we can also include the "fetch" time, or the time required for "access." For example, one time unit may be assigned to the value of "a" for getting ready for the next operation. In this manner, we can schedule both the "instruction" and the "execution" phases.

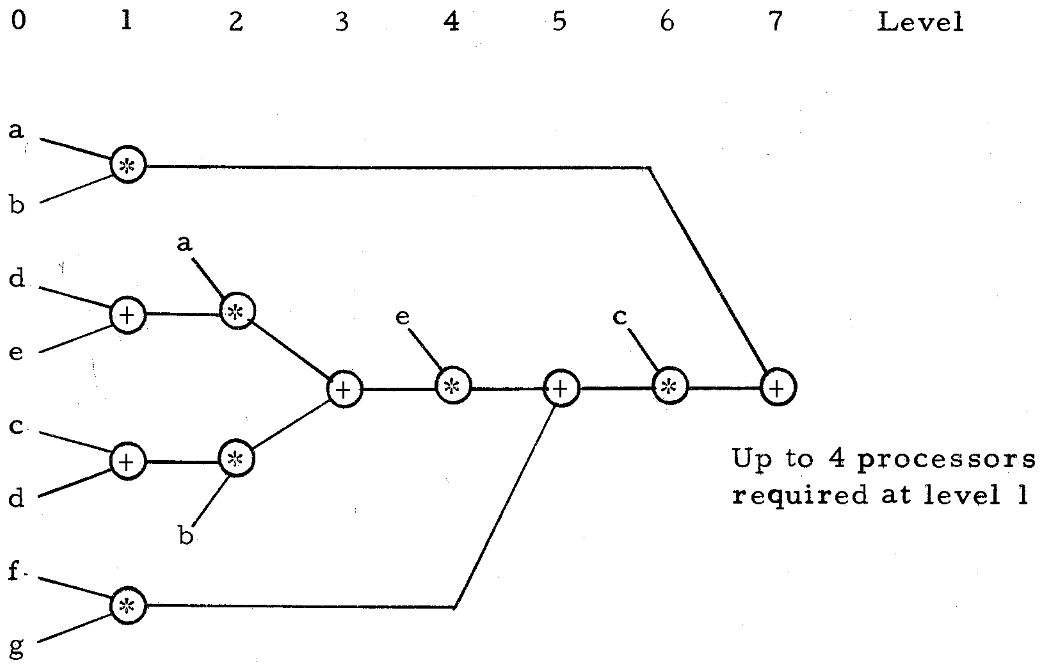


Figure 8. 8. a. Earliest-stage-assignment.

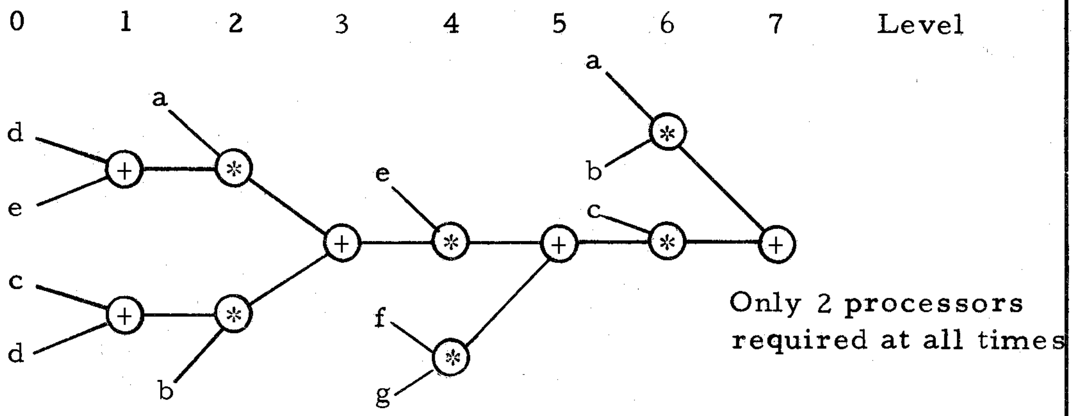


Figure 8. 8. b. Latest-stage-assignment.

Internal Tableau Operations

Signal flowgraph techniques. One of the major advantages of our tableau method is that the flowgraph compatibility built in the method (Chapter III) allows us to make a full use of available signal flowgraph techniques. Any operation that can be performed on a signal flowgraph can be performed on a tableau as well. Let us present a simple but well known example of economic equilibrium between the rate of investment I, and the savings S (Hirsch, 1966). Figure 8.9 should be self-explanatory.

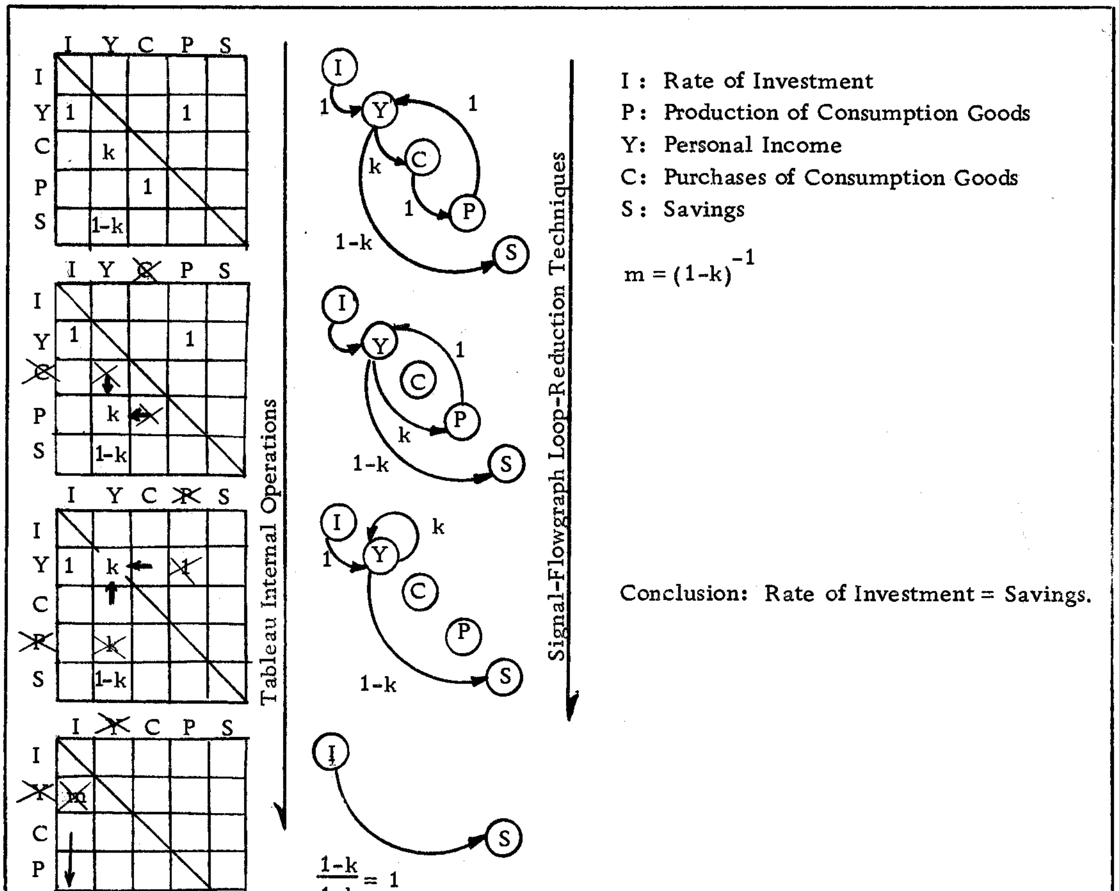


Figure 8.9. Signal flowgraph example.

Simplex Example 8. 2. Restaurant problem. A man has just purchased a restaurant with 18 tables and 60 chairs. He has the choice of setting one table with two chairs (expected profit \$4 per hour), one table with four chairs (\$8 per hour), two tables with four chairs (\$10 per hour), or two tables with six chairs (\$14 per hour). How many of which arrangements should he have to maximize his profit?

Solution to Example 8. 2. Figure 8. 10 shows how this problem could be solved by the internal manipulation of the tableau.

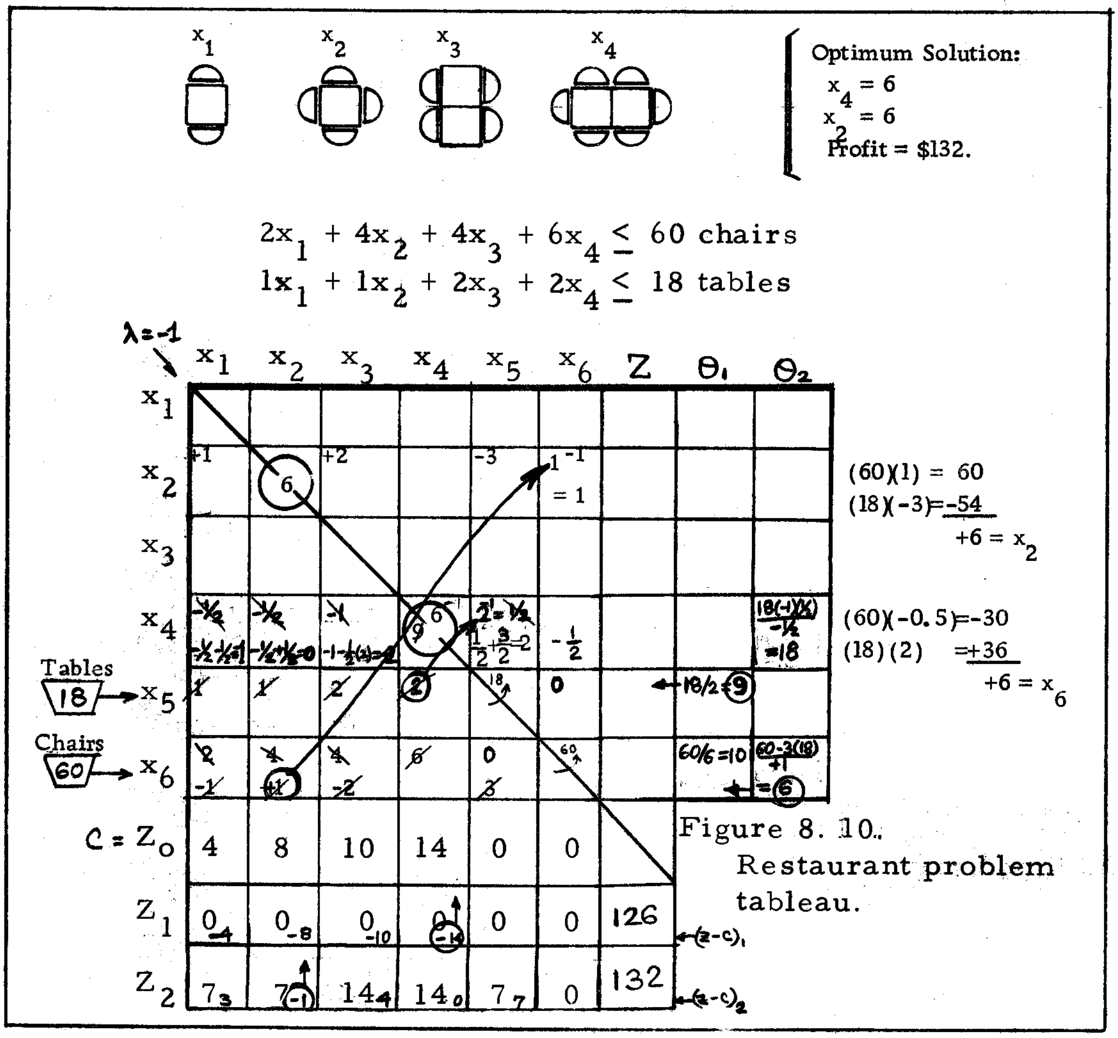


Figure 8. 10. Restaurant problem tableau.

Future

What we have said is enough to show that the three subjects, communication, organization, and control are linked in so many ways that the discussion of one of them inevitably brings in the two others. Randomness, errors, information, meaning, value, which we have mentioned, are important and correlated ideas. The new field of engineering that embraces them all, called 'systems engineering,' has absorbed a number of earlier technical developments, such as circuit theory or those connected with the design and utilization of automatic computers. It is still too early to say where systems engineering will lead; while it was originally an engineering technique, it is now being applied to problems arising in the life sciences, the social sciences, and even, in some cases, the humanities. Other developments concern logic and philosophy (Le Corbeiller, 1963).

Future Areas of Research

Considerable amount of work is needed in practically every subject we have treated in this treatise, and undoubtedly there are many areas we have not touched which will be vitally important in the future of Systems Theory.

Communication. If we are forced into making the choice of area most imminently needing of research, we will not hesitate to choose communication and information. The first reason is obviously the ever-increasing need for a universally acceptable "language" for systems study. Today's complex systems require specialists from various backgrounds to work together on a problem, and this is becoming increasingly difficult. Of course, this is precisely the reason why the work on this particular treatise was undertaken.

The second reason is, simply, because everything we do is a communication of one form or another. Not only will the study of communication render our actions more efficient and rational, but it will help us to understand our own nature. Perhaps it is not so surprising that our nerve system acts according to principles discovered in Information Theory. After all, it is a communication system in its own right. But what is surprising is that we seem to have been coded according to the principles we have uncovered in Information Theory. According to the Nobel Laureate George W. Beadle, we are the product of information carried by DNA (Deoxyribonucleic acid) and transmitted by the messenger RNA (ribonucleic acid) to the ribosomes in the cytoplasm, for the specific synthesis of protein.

> DNA may be described as a four-symbol language, while, in the same terms, proteins are 'written' in a twenty-symbol language. How is one translated into the other? One possibility, considered early in decoding game, is that 'three-letter words' in DNA, of which there are sixty-four possibles if the molecule is read in one direction only, somehow specify single amino acids. Substantial experimental evidence is now available supporting the hypothesis of protein synthesis...(Beadle, 1963).

It is even hypothesized that our evolution is the result of mutation in the transmission of information due to "noise."

All these point out to us engineers, that perhaps we should turn our attention to the study of "natural" systems in trying to find solutions to some of our most complex systems problems. For instance, the problem of "adaptive control" or "learning machine" may find its answer in deliberately bringing in randomness (a heuristic approach rather than deterministic).

The problem of utility may likewise find its solution in natural systems. How does an organism show a preference? Does it have a cardinal utility function or only an ordinal one?, etc...

Organization. From a purely engineering point of view, a more deterministic approach may be easier at first. From this standpoint, the use of a tableau or a network may provide a suitable framework in the study of problems involving stochastic properties. We have so far dealt with decision-making problems in which the process was Markov in nature. Obviously, the next step will be to add a new dimension. This new dimension may be brought in by having several decision criteria, by having transition probabilities depend on "memory," or by bringing in "noisy" or "lossy" decision criteria, and so on. In other words, the added dimension in an added degree of freedom in the corresponding tableau which may be manifested physically, time-wise, frequency-wise (probability is an expression of frequency), state-wise, or phase-wise. As far as a tableau is concerned, it can accomodate any of these dimensions with equal ease. Ronald Howard conceived his dynamic programming as a three dimensional model (present state, alternatives, succeeding states: 1960, p. 33). This imposed an artificial restriction that made dynamic programming problems to be one level lower than the recursive games proposed by Luce and Raiffa (1957, p. 461). A tableau, as we have seen from Chapter VI, provides an excellent representation for tracing back the "memory" or to figure out the future strategies to be taken by opponents in a game.

Control. The problem of control is probably the most difficult of all three fields. In order to control, we need a criterium. In all likelihood, some sort of utility function will have to be used. The application of Bayes' rule in estimating uncontrollable inputs should be investigated also. We have the suspicion that the use of transforms may be very profitable in both cases. The amount of control which can be exerted on a model, may perhaps be investigated by considering the information capacity of the model as a channel. And this brings us back to our starting point: the investigation of information and communication theory problems.

Systems Theory

The study of a system became essentially that of communication, organization, and control of the system. The communication phase corresponds to receiving information, or analyzing the system. The organization is obviously the synthesis phase of the systems theory. The control is the decision-making process concerning the state of the system. This is indeed what we have proposed in the start of our study.

BIBLIOGRAPHY

Arbib, Michael A. 1964. Brains, machines, and mathematics. New York, McGraw-Hill. 152 p.

Ash, Robert B. 1965. Information theory. New York, Interscience. 339 p.

Ashby, W. Ross. 1963. An introduction to cybernetics. Science ed, New York, Wiley. 295 p.

Avondo-Bodino, Giuseppe. 1962. Economic applications of the theory of graphs. New York, Gordon and Breach. 111 p.

Beer, Stafford. 1964. Cybernetics and management. Science ed., New York, Wiley. 214 p.

Bell, D. A. 1962. Intelligent machines. New York, Blaisdell. 98 p.

Bellman, Richard. 1953. An introduction to the theory of dynamic programming. Santa Monica, Calif., Rand. 99 p.

________________. 1957. Dynamics programming. Princeton, New Jersey, Princton University Press. 342 p.

Bellman, Richard E. and Stuart E. Dreyfus. 1962. Applied dynamic programming. Princeton, New Jersey, Princeton University Press. 363 p.

Bellman, Richard and Robert Kalaba. 1965. Dynamic programming and modern control theory. New York, Academic Press. 112 p.

Berge, Claude. 1962. The theory of graphs and its applications. London, Methuen. 247 p. (Translated from French)

Bonini, Charles P. 1963. Simulation of information and decision systems in the firm. Englewood Cliffs, N. J., Prentice-Hall. 160 p.

Bradley, Joseph F. 1964. Administrative financial management. New York, Holt. 612 p.

Baun, Edward L. 1963. Digital computer design. New York, Academic Press. 606 p.

Chamberlain, Howard Emil. 1965. Estimating highlead logging performances through statistical models. Master's thesis. Corvallis, Oregon State University, 74 numb. leaves.

Chernoff, Herman and Lincoln E. Moses. 1959. Elementary decision theory. New York, Wiley. 364 p.

Chestnut, Harold. 1965. Systems engineering tools. New York, Wiley. 646 p.

Chow, Yutze and Etienne Cassignol. 1962. Linear signal-flow graphs and applications. New York, Wiley. 144 p.

Chu, Yaohan. 1962. Digital computer design fundamentals. New York, McGraw-Hill. 481 p.

Compton, J. B. and D. E. Moody. 1965. Flowgraphing executive problems. IEEE Transactions on Engineering Management EM-12:143-149.

Control Engineering. Second manual of digital techniques. New York, McGraw-Hill. 125 p.

Courant, R. and D. Hilbert. 1953. Methods of mathematical physics. Vol. 1. New York, Interscience. 561 p.

Cox, D. R. and Walter L. Smith. 1961. Queues. London, Methuen. 180 p.

Dantzig, George B. 1963. Linear programming and extensions. Princeton, N. J., Princeton University Press. 625 p.

Dantzig, George B. and David L. Johnson. 1964. Maximum payloads per unit time delivered through an air network. Operations Research 12:230-236.

Davenport, Wilbur B. Jr. and William L. Root. 1958. Random signals and noise. New York, McGraw Hill. 393 p.

Denardo, Eric V. 1966. Dynamic programming. Lecture delivered in: Modern systems theory and its applications to large scale systems: seminar held at University of California, Los Angeles.

Dervitsiotis, Kostas Nicholaus. 1965. Critical path budgeting: a synthesis of budgeting and critical path scheduling. Master's thesis. Corvallis, Oregon State University, 137 numb. leaves.

Desmonde, William H. 1964. Computer and their uses. Englewood Cliffs, N. J., Prentice-Hall. 296 p.

Doebelin, E. O. 1962. Dynamic analysis and feedback control. New York, McGraw-Hill. 401 p.

Dreyfus, Stuart E. 1965. Dynamic programming and the calculus of variations. Santa Monica, Calif., Rand. 511 p.

Evan, William M. 1965. Toward a theory of inter-organizational relations. Management Science 11:B-217-B-230.

Farrar, Donald E. 1962. The investment decision under uncertainty. Englewood Cliffs, N. J., Prentice-Hall. 90 p.

Flagle, Charles D. et al. (eds.) 1960. Operations research and systems engineering. Baltimore, Johns Hopkins Press. 889 p.

Ford, L. R., Jr. and D. R. Fulkerson. 1962. Flows in networks. Princeton, N. J., Princeton University Press. 194 p.

Forrester, Jay W. 1964. Common foundations underlying engineering and management. IEEE Spectrum 1(9):66-77.

Forster, Michael. 1966. Dynamics of economic systems. Lecture delivered in: Modern systems theory and its applications to large scale systems: seminar held at University of California, Los Angeles.

Fraser, D. A. S. 1957. Nonparametric methods in statistics. New York, Wiley. 299 p.

Freeman, Herbert. 1965. Discrete time systems. New York, Wiley. 241 p.

Galamoos, Robert. 1962. Nerves and muscles. New York, Doubleday. 159 p.

Gale, David. 1960. The theory of linear economic models. New York, McGraw-Hill. 330 p.

Gibson, John E. 1963. Nonlinear automatic control. New York, McGraw-Hill. 585 p.

Goddard, L. S. 1963. Mathematical techniques of operation research. Oxford, Pergamon Press. 230 p.

Goldstein, Herbert. 1950. Classical mechanics. Reading, Addison-Wesley. 399 p.

Gordon, Richard Allen. 1966. Analysis of human and equipment performance in critical path scheduling. Master's thesis. Corvallis, Oregon State University, 87 numb. leaves.

Graves, Robert L. and Philip Wolfe (eds.). 1963. Recent advances in mathematical programming. New York, McGraw-Hill. 347 p.

Greenberger, Martin (ed.). 1962. Computers and the world of the future. Cambridge, M.I.T. Press. 340 p.

Greene, James H. 1965. Production control systems and decisions. Homewood, Richard D. Irwin. 605 p.

Greeniewski, Henryk. 1965. Intention and performance: a primer to cybernetics of planning. Management Science 11:763-782.

Hadley, G. 1961. Linear algebra. Reading, Pa., Addison-Wesley. 290 p.

________________. 1962. Linear programming. Reading, Pa., Addison-Wesley. 520 p.

________________. 1964. Nonlinear and dynamic programming. Reading, Pa., Addison-Wesley. 484 p.

Hall, Arthur D. 1962. A methodology for systems engineering. Princeton, N.J., Nostrand. 478 p.

Halmos, Paul R. 1958. Finite-dimensional vector spaces. 2nd ed. Princeton, N. J., Nostrand. 200 p.

________________. 1960. Naive set theory. Princeton, N. J., Nostrand. 104 p.

Hammond, P. H. 1958. Feedback theory and its applications. New York, Macmillan. 348 p.

Hammond. S. B. 1961. Electrical engineering. New York, McGraw-Hill. 566 p.

Hellerman, H. 1966. Parallel processing of algebraic expression. IEEE Transactions on Electronic Computers EC-15:82-91.

Henderson, James M. and Richard E. Quandt. 1958. Microeconomic theory. New York, McGraw-Hill. 291 p.

Hespos, Richard F. and Paul A. Strassmann. 1965. Stochastic decision trees for the analysis of investment decisions. Management Science 11:B-244-B-259.

Howard, Ronald A. 1965. Bayesian decision models for systems engineering. IEEE Transactions on Systems Science and Cybernetics SSC-1:36-40.

_______________. 1960. Dynamic programming and Markov processes. New York, Wiley. 136 p.

Huggins, William Herbert. 1942. Radio influence characteristics of high voltage electrical discharges. Master's thesis. Oregon State University. 145 numb. leaves.

Iijima, Taizo. 1966. The principles of pattern recognition. Electronics (Japan) 11:259-270.

Inoue, Michael Shigeru. 1964. Critical path scheduling: tableau method. Master's thesis. Corvallis, Oregon State University, 140 numb. leaves.

Inoue, Michael S. and Louis N. Stone. 1965. Lecture notes on introduction to electronic data processing: hardware, software, and system. Summer course given to employees of Oregon State Department of Employment, Salem, Oregon, 241 numb. leaves. (To be published by McGraw-Hill in 1967)

Intrilligator, Michael. 1966. Application of control theory to problems in economic allocation. Lecture delivered in: Modern systems theory and its applications to large scale systems: seminar held at University of California, Los Angeles.

Iverson, Kenneth E. 1962. A programming language. New York, Wiley. 286 p.

Johnston, John B., G. Baley Price and Fred S. Van Vleck. 1966. Linear equations and matrices. Reading, Pa., Addison-Wesley. 308 p.

Karplus, Walter J. 1958. Analog simulation. New York, McGraw-Hill. 434 p.

Kennedy, John F. 1961. State of union message. On the outgoing administration. Washington, D. C. January 31, 1961.

Kipiniak, Walerian. 1961. Dynamic optimization and control. Cambridge, Massachusetts Institute of Technology Press. 233 p.

Kuo, Benjamin C. 1963. Analysis and synthesis of sampled-data control systems. Englewood Cliffs, N. J., Prentice-Hall. 528 p.

Le Corbeiller, Philippe. (ed.) 1963. The language of science. New York, Fawcett. 176 p.

Leitmann, George (ed.) 1962. Optimization techniques with applications to aerospace systems. New York, Academic Press. 453 p.

Levin, Richard I. and C. A. Kirkpatrick. 1965. Quantitative approaches to management. New York, McGraw-Hill. 365 p.

Lockheed Missiles and Space Company. 1965. California statewide information system study. Sunnyvale, California. 228 p.

Luce, R. Duncan and Howard Raiffa. 1957. Games and decisions: introduction and critical survey. New York, Wiley. 509 p.

Lukasiewicz, Jan. 1963. Elements of mathematical logic. New York, McMillan. 124 p. (Translated from Polish by Olgierd Wojtasiewicz)

Magee, John F. 1958. Production planning and inventory control. New York, McGraw-Hill. 333 p.

Mandl, F. 1957. Quantum mechanics. 2d ed. New York, Academic Press. 267 p.

Markov, A. A. 1962. Theory of algorithms. 2d ed. Jerusalem, Israel Program for Scientific Translations. 444 p. (Translated from Russian by Jacques J. Schorr-Kon and PST staff)

Massachusetts Institute of Technology. Operations Research Center. 1959. Notes on operations research. Cambridge. 256 p.

McKenna, Joseph P. 1958. Economic theory. Holt, Reinhart and Winston. 319 p.

McMillan, Claude and Richard F. Gonzalez. 1965. Systems analysis: a computer approach to decision models. Homewood, Ill., Richard D. Irwin. 336 p.

Meyer, W. Lyle. 1966. Use of critical path method simplified by introduction of circle notation, overlap feature. SAE Journal 74(3):80-83.

Middleton, David. 1960. An introduction to statistical communication theory. New York, McGraw-Hill. 1140 p.

Miernyk, William H. 1965. Input-output analysis. New York, Random. 156 p.

Mood, Alexander M. and Franklin A. Graybill. 1963. Introduction to the theory of statistics. 2d ed. New York, McGraw-Hill. 443 p.

Moose, Victor Corson. 1966. Design, construction, and evaluation of a mechanical queueing simulator. Master's thesis. Corvallis, Oregon State University, 71 numb. leaves.

Mostow, D. George, Joseph H. Simpson and Jean-Pierre Meyer. 1963. Fundamental structures of algebra. New York, McGraw-Hill. 585 p.

Nagel, Ernest and James R. Newman. 1958. Gödel's proof. New York, New York University Press. 118 p.

Owen, George E. 1961. Fundamentals of scientific mathematics. Baltimore, Johns Hopkins Press. 274 p.

________________. 1963. Introduction to electromagnetic theory. Boston, Allyn and Bacon. 506 p.

Parzen, Emanuel. 1960. Modern probability theory and its applications. New York, Wiley. 464 p.

Pierce, John F., Jr. 1964. Some large-scale production scheduling problems in the paper industry. Englewood Cliffs, N. J., Prentice-Hall. 255 p.

Raisbeck, Gordon. 1963. Information theory. Cambridge, M.I.T. Press. 105 p.

Riggs, James Lear. 1963. Influence of environmental factors on human mental and motor performance in console operations. Ph.D. thesis. Oregon State University. 88 numb. leaves.

Riggs, James L. and Charles O. Heath. 1963. A working manual for critical path scheduling. 2d ed. Corvallis, Oregon, Hares. 137 p.

________________. 1966. A management guide to cost reduction through critical path scheduling. Englewood Cliffs, N. J., Prentice-Hall. n. p.

Riggs, James L. and Michael S. Inoue. 1965. Critical path scheduling: tableau method. Corvallis, Oregon State University, Engineering Experiment Station. 25 p.

Rojansky, Vladimir. 1938. Introductory quantum mechanics. Englewood Cliffs, N.J., Prentice-Hall. 544 p.

Sasieni, Maurice et al. 1959. Operations research methods and problems. New York, Wiley. 316 p.

Savage, Jay M. 1963. Evolution. New York, Holt. 126 p.

Schlaifer, Robert. 1959. Probability and statistics for business decisions. New York, McGraw-Hill. 732 p.

Schwartz, M. D. 1966. Matrices and linear system theory. In: Modern systems theory and its applications to large scale systems: seminar held at University of California, Los Angeles, UCLA, 48 numb. leaves.

Sengupta, S. S. and R. L. Ackoff. 1965. Systems theory from an operations research point of view. IEEE Transactions on Systems Science and Cybernetics SSC-1:9-13.

Seshu, Sundaram and Myril B. Reed. 1961. Linear graph and electrical networks. Reading, Addison-Wesley. 315 p.

Shannon, Claude E. and Warren Weaver. 1963. The mathematical theory of communication. Urbana, The University of Illinois Press. 117 p.

Sherwin, Chalmers W. 1959. Introduction to quantum mechanics. New York, Holt. 385 p.

Starr, Martin Kenneth. 1964. Production management. Englewood Cliffs, N. J., Prentice-Hall. 533 p.

Thuesen, H. G. and W. J. Fabrycky. 1964. Engineering economy. 3d ed. Englewood Cliffs, N. J., Prentice-Hall. 525 p.

Tonge, Fred M. 1961. A heuristic program for assembly line balancing. Englewood Cliffs, N. J., Prentice-Hall. 115 p.

Trakhtenbrot, B. A. 1963. Algorithems and automatic computing machines. Boston, Heath. 101 p. (Translated from Russian)

Vajda, S. 1961. Mathematical programming. Reading, Addison-Wesley. 310 p.

________________. 1962. Readings in mathematical programming. 2d ed. New York, Wiley. 130 p.

Van Valkenburg, M. E. 1960. Introduction to network synthesis. New York, Wiley. 498 p.

Veinott, Arthur F. Jr. (ed.) 1965. Mathematical studies in management science. New York, Macmillan. 481 p.

Von Neumann, John. 1963. The computer and the brain. New Haven, Conn., Yale University Press. 82 p.

Von Neumann, John and Oskar Morgenstern. 1964. Theory of games and economic behavior. Science ed. New York, Wiley. 641 p.

Wade, Thomas L. 1951. The algebra of vectors and matrices. Reading, Massachusetts, Addison-Wesley. 189 p.

Wald, Abraham. 1950. Statistical decision functions. New York, Wiley. 179 p.

Warfield, John N. 1959. Introduction to electronic analog computers. Englewood Cliffs, N. J., Prentice-Hall. 175 p.

Wiener, Norbert. 1961. Cybernetics or control and communication in the animal and the machine. 2d ed. Cambridge, Massachusetts Institute of Technology Press. 212 p.

Wolfe, Philip. 1966. Nonlinear programming. Lecture delivered in: Modern systems theory and its applications to large scale systems: seminar held at University of California, Los Angeles.

Wolff, Ronald. 1966. Lecture notes in queueing. Berkeley, University of California. 67 numb. leaves.