

THE UNIVERSITY OF MICHIGAN

Technical Report 21

On the Representation of Markovian Systems by Network Models

Victor L. Wallace

CONCOMP: Research in Conversational Use of Computers
F. H. Westervelt, Project Director
ORA Project 07449

supported by:

ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE
WASHINGTON, D. C.

CONTRACT NO. DA-49-083 OSA-3050
ARPA ORDER NO. 716

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

August 1969

Abstract

Formal, unambiguous mathematical structures are developed for representing Markovian queueing networks and for systematically constructing a description of a continuous-parameter Markov chain model from a description of the network diagram. A formal queueing diagram notation is developed as a pictorial language. An approach to the problem of decomposition and recomposition of Markovian queueing networks is presented, and applied to realistic queueing networks.

Table of Contents

	<u>Page</u>
Preface	vii
List of Figures	ix
List of Tables	xi
1. Introduction	1
2. A Syntax of the Pictorial Language	5
3. Semantics of the Pictorial Language	11
4. Defining a Network Model	27
5. Element Representation	29
6. The Network as a Markov Chain	53
7. The Translation of Networks to Equilibrium Equations	79
References	112

Preface

This report treats a general approach to the decomposition and recomposition of Markovian queueing networks. This theory was developed in 1967 as part of a project aiming at providing a computer program to produce transition intensity matrices from a description of the network in block-diagram form automatically by use of a digital computer.

The system which has finally been implemented [6, 7] under this project is based upon a gross simplification of the notions presented here, with an attendant reduction in the generality of the networks which may be so treated. Indeed, because of the considerable difficulties which a direct implementation of this theory in such a programming system entails, the simplified theory assumes a distinctly different form.

Nevertheless, this report should be of sufficient interest to its audience both as a progenitor of the notions finally implemented, and as a theoretical development on its own, capable of further theoretical development as part of the long neglected field of queueing networks.

V. L. Wallace
Ann Arbor

List of Figures

	<u>Page</u>
2.1 A network diagram.	6
3.1 A "holding server."	14
3.2 Schematic of port variables.	23
5.1 The arrival element.	31
5.2 The exponential server.	32
5.3 The queue.	39
5.4 The random switch.	44
6.1 Illustrating the "propagation of an epoch."	65
6.2 An epoch graph fragment corresponding to Fig. 6.1.	66
6.3 Graphs related to the epoch graph.	72
6.4 An epoch graph.	76
7.1 Illustrating two cases for consolidation.	80
7.2 Illustrating the "consolidated" networks for Fig. 7.1.	82
7.3 Schematic of port variables.	85
7.4 Relationships of port variables in connected elements.	85
7.5 Illustrating epoch-splitting.	87
7.6 Illustrating epochs and their influence.	89
7.7 Illustrating epochs and their influence.	92
7.8 Illustrating externally determined epochs and their influence.	98
7.9 Queue-and-server consolidation.	101

List of Tables

		<u>Page</u>
I	Semantics of State Variables, Parameters, and Auto-Epochs	17
II	Semantics of Input and Output Port Variables	21
III	Semantics of Contents Sensing and Value Collecting Ports	22
IV	Element Definitions	50

1. INTRODUCTION

The Markovian systems of the title are, by definition, systems whose behavior is modeled by a continuous--parameter Markov chain with stationary transition probabilities.¹ A network model, on the other hand, is a model consisting of distinct elements, interconnected through well defined interfaces. Such a model is often represented compactly in the form of a network diagram, consisting of separate elementary symbols and lines between them. The three representations—network diagram, network model, and Markov chain—can be used as three alternative descriptions of the same object, the Markovian system. That is, they can be used as alternative descriptions provided that the network diagram and the network model are formally and unambiguously defined. If they are, then it should also be possible to translate a description in any one of these forms into an equivalent description in any other.

It is the purpose of this report to develop formal and unambiguous mathematical structures for both the network diagrams and the network models, and to show how a diagram can be translated into an equivalent Markov chain through intermediate use of these structures. The network model structure will be defined in a form specifically tailored to make this translation convenient. It will represent an algebraic description of the meaning behind the symbols used in the network diagram. The network diagram structure will be defined so that the diagrams will most flexibly and unambiguously approximate the crude, ambiguous (but natural) diagrammatic

forms used by systems analysts and theorists as visual aids. In this way, the diagrams will offer maximum facility to these people as a means of communication.

This work represents a first stage in development of the theoretical machinery required for a graphic man-computer system for Markovian systems analysis. The areas of potential application of this system are potentially many, and include computer systems, communications switching systems, traffic control systems, assembly lines, and reliability studies. (The class of Markovian systems includes all those ordinarily called Markovian queueing systems.) While analysis of a Markov chain can proceed by analytical, numerical, or simulational means, this man-computer system will utilize a numerical analysis procedure based on that of the Recursive Queue Analyzer² in order to achieve a good compromise between response time and generality. This is an algorithm which requires only that the matrix of transition intensities of the Markov chain (the coefficient matrix of the Kolmogorov differential equation) be known. Thus this matrix, together with information identifying states, will be considered the basic form, descriptive of the Markov chain.

This report has three principal parts. Sections 2 and 3 pose a legitimate picture language format for Markov network diagrams, and abstract a framework for the transfer of meaning from the network diagrams to the algebraic network model. Sections 4 through 6 develop the algebraic model and treat its relationship to the Markov chain. Because the algebraic model

is relatively complicated and abstract, its development is carried through a series of trial models showing, by counterexample, why each new complication is a necessary improvement. Section 7 treats an operation called consolidation and the procedure for translation from the primitive network model to the transition intensity matrix describing the Markov chain.

2. A SYNTAX OF THE PICTORIAL LANGUAGE

2.1 PICTORIAL LANGUAGES

Formal representations of Markovian systems in diagram form are actually expressions in a graphical language. More precisely, it is two-dimensional pictorial language, to use W. R. Sutherland's³ term. It is a means of conveying precise meaning through pictorial symbols and syntax. Because the representations are descriptive of a problem to be solved rather than of the means of solution, such a language is also called a problem-oriented pictorial language. The chief utility of pictorial languages arises from their compactness and from the high degree of instant recognition possible compared with textual languages.

2.1.1 An Example

Consider, for example, the diagram of Figure 2.1. With suitable definitions, the symbols and syntax of this diagram could have the following meaning: Poisson arrivals of tasks, having mean rate of occurrence λ , are accepted into a queue, with up to a maximum of n_1 tasks present in the queue at any one time. A server removes tasks from the queue and processes them for independent, exponentially distributed intervals of time having mean $1/\mu_1$. Whenever the two other queues have less than a total of n_5 tasks present at the time of a completion, the completed task is randomly assigned to the upper queue (with probability p) or the lower queue (with probability $1-p$). Otherwise the service is restarted and the server held for

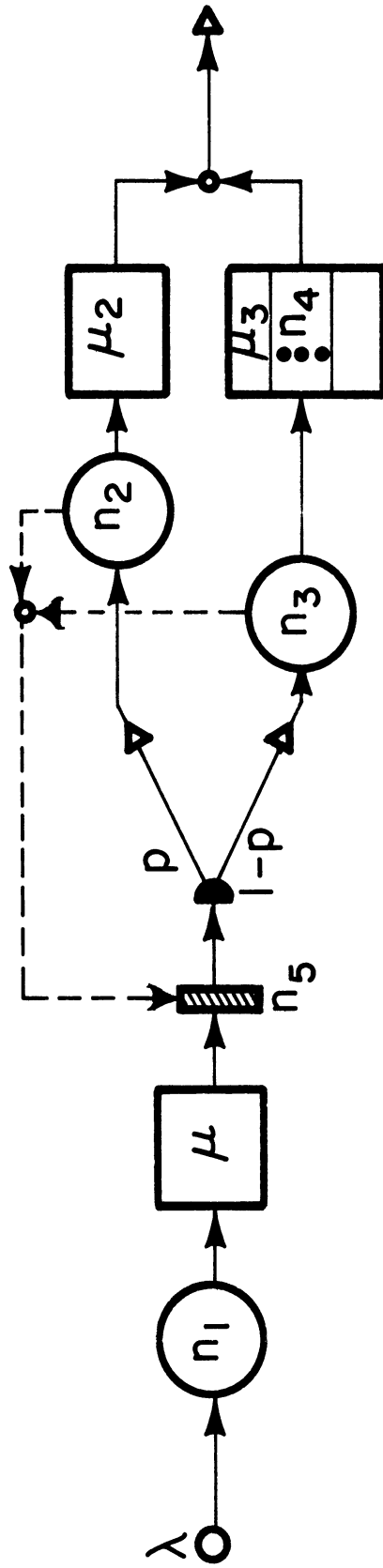


Figure 2.1 A network diagram.

another exponentially distributed interval of time. Tasks assigned to the upper queue are removed from the queue and processed, one at a time, by a server which is held for an independent exponentially distributed period of time having mean $1/\mu_2$. Upon completion of this holding, the task exits from the system and a new task is immediately drawn from the upper queue, if the queue is not empty. Tasks assigned to the lower queue go to any of n_4 servers whenever one of them is vacant. These servers are each held for independent exponentially distributed intervals with mean $1/\mu_3$. Every completion produces an exit of a task from the system, and a new task is immediately drawn from the lower queue into the vacated server, if the queue is not empty.

2.1.2 The Need for a New Language

That, briefly, is the meaning which can be assigned to the diagram of Figure 2.1. In some senses, that description is incomplete because of the difficulty of systematically conveying all possible conditions and actions in English prose (i. e., textually). It will be seen later than any questions arising from the ambiguities of the description can be answered when the symbols are adequately defined. The main issue here is that the diagram is clearly a means of description which is superior to prose. It is compact and easy to visualize.

At present, no generally accepted pictorial language for the purpose of describing Markovian systems exists. As used by queueing theorists and analysts, diagrams are used as discussion aids rather than as complete

symbolic representations. It is not uncommon, for example, for textbooks to use the same diagram for two different systems—the difference being drawn out in the discussion. Thus, these diagrams are usually ambiguous and can't be formally used as prototypes for a picture language. The block-diagram representations used by some simulator programs⁴ are unambiguous, but lack both the conciseness and the flexibility desired. They also are more general than necessary, since they can usually be used to describe many nonMarkovian systems.

It is necessary, therefore, to define a pictorial language here which can be used for the purpose. However, rather than define a particular language, a class of languages which differ from one another only in their vocabulary will be defined. In other words, it will not be assumed that fixed set of symbols will be used. Rather, only the means of defining the symbols, and the form of the basic construction of expressions will be prescribed. In this way a "living language" is constructed. Of more practical significance, conciseness is preserved over wide ranges of application when special purpose symbols and syntactic constructions can be invented by a user as needed. This will obviate much of the awkwardness found in the simulation languages, and permit the abbreviation of frequently used constructions.

2.2 A PROPOSED SYNTAX

A network diagram (which is an expression in the picture language) will be characterized by a set of distinct symbols with lines between them.

Each symbol has a characteristic shape and a number of special connection points physically located on it. Each connecting point is the end-point of at most one connecting line, and has a specific orientation on the symbol, which identifies it. (Points in Figure 2.1 which appear to have more than one connecting line are actually small symbols with more than one connecting point.) Each connecting line has a type (dotted, solid, etc.) and joins exactly two connecting points.

The connecting points also have types, which determine what connections can legally be made. There are four types in use in Figure 2.1:

- 1) task transfer-input;
- 2) task transfer-output;
- 3) contents sensing; and
- 4) value collecting.

Others are possible, and this is one of the syntactic variables of the language. In the language of Figure 2.1, connections are only permitted between a point of type (1) and a point of type (2), or between a point of type (3) and a point of type (4). The former connection is shown solid, the latter dotted. For simplicity, only this connection syntax will be used in this report.

Finally, each symbol has a set of locations for parameters. The orientation of the parameter identifies it and its range (integer, real, etc.). The parameter may be supplied as any algebraic expression of appropriate range.

It will be evident in the next section that the connection syntax is irrelevant to the algebraic model or its translator. It provides only a protection against unintended constructions. An algebraic meaning for the connecting line will always exist, regardless of the point-types involved, but it may represent nonsense if the connection syntax is invalid.

3. SEMANTICS OF THE PICTORIAL LANGUAGE

Having established a syntax, which defines the form which Markovian network diagrams may legally take, the meaning which the diagram conveys must be described. This meaning can be awkwardly conveyed in English, as the description of Figure 2.1 testifies. However, our purpose is to describe this meaning in a formal algebraic form, to enable systematic transfer of this meaning to the description of a Markov chain. In this sense, set and function concepts are used as a meta-language with which to define the picture language expressions. The English descriptions will be used to gain insight into the forms which the algebraic descriptions must take.

3.1 AN ASSUMPTION—MEANING INDEPENDENT OF CONTEXT

One important property of the language which must be assumed is that each symbol have a definable meaning independent of its position (or context) in a network diagram. It will be seen in this section that each of the symbols used in Figure 2.1, as well as many others which can be invented, can be so described in English. Since this is the case, our plan to seek to do so also in formal algebraic terms appears reasonable.

In order to distinguish between language elements and the things they represent, it will be noticed that separate terms are used. Thus diagrams represent networks, whereas symbols represent elements, connecting lines represent connections, and connection points represent ports. (The phrase "The meaning of a language element" is synonymous to the phrase "a

description of the thing represented by the language element.")

3.1.1 An Example

To give an example of a verbal, context-independent definition of a symbol, consider the following definition of a queue. The queue has an input port, an output port, and a contents sensing port. (For simplicity, the operation of the contents-sensing port will not be treated here.) It has a parameter, represented here by the symbol N , which indicates the maximum number of waiting tasks which can be allowed in the queue. Let the number of waiting tasks be represented here by s . This queue is affected whenever tasks are offered by the element connected at its input. If y_1 tasks are offered at the input, and x_2 tasks are acceptable to the element connected at the output, then $s+y_1$ tasks will be offered at the output. The number of tasks actually sent to the output will be the smaller of $s+y_1$ and x_2 . The number of waiting tasks is changed to N if $s+y_1-x_2 > N$; to zero if $s+y_1-x_2 < 0$; and to $s+y_1-x_2$ otherwise. The number of tasks acceptable to the queue, at its input, is $N-s+x_2$. The number actually taken at the input is the smaller of $N-s+x_2$ and y_1 .

The queue is also affected whenever tasks are requested by the element connected at its output port. If y_2 tasks are requested at the output, and x_1 tasks are available from the element connected at the input, then y_2+N-s tasks will be requested by the queue from the element connected at its input. The number of tasks actually taken at the input will be the smaller of y_2+N-s and x_1 . The number of waiting tasks is changed to N if

$s+x_1-y_2 > N$; to zero if $s+x_1-y_2 < 0$; and to $s+x_1-y_2$ otherwise. The number of tasks sent to the output is $s+x_1$ if $s+x_1-y_2 \leq 0$; and y_2 if $s+x_1-y_2 > 0$.

3.2 A FRAMEWORK FOR SEMANTIC DESCRIPTION OF SYMBOLS

The verbal definition of the queue is very wordy and somewhat repetitious. The verbal description of the meaning of Figure 2.1 was also. The key to reducing this wordiness, or at least putting it in some perspective, lies in attempting to observe the entities which all elements have in common. Then, by naming these entities, the cumbersome phrases which describe them can be replaced by something more technical.

For example, the definition of the queue made frequent reference to "the element connected to (a port)." Such a reference is to be expected in the definition of perhaps every element which has a port, unless the elements related by the connection are truly independent of one another. Thus, the term associate will be used to refer to an element connected to the element under study. The term associate at (a port) can be used when specification of the port is necessary.

3.2.1 State Variables

Less trivially, the phrase "the number of tasks waiting" describes the status of the element at any particular time. It is to be expected that there will be similar status variables for other elements, like "the number of tasks in service" for a multiple server element, or "the current phase of service" for servers with Erlang distributed holding times.⁵ Such

variables, when they are present in an element, will be referred to as state variables of the element, or sometimes as state components (for reasons that will be clear in the next section). There may be more than one such variable for a single element, for combinations of element should also be legal elements of a system. To illustrate, Figure 3.1 demonstrates a possible equivalence, in diagram form, between a "holding server" and a queue and server connected together. The state variables might well be jointly the "number waiting" and the "number in service."

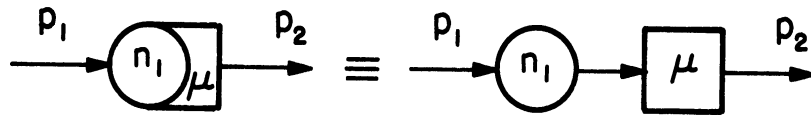


Figure 3.1 A "holding server."

The range of values which the state variable of an element can assume is a set which is distinctive of the element. We call this set S the state set of the element. It is also a property of the element, like the set P of ports of the element. That is, an element e consists of a set P , a set S , and other things. The members of S are either integers or integer vectors, depending on whether there is one, or more than one, state variable to the element.

Table I lists a set of useful symbols for a pictorial language for Markovian networks, along with their names, the semantics associated with

their state variables, their state sets, and some other properties which will be described in subsections 3. 2. 2 and 3. 2. 3. A verbal description of the meaning of some of these symbols will be given in section 3. 3 and in section 5.

3. 2. 2 Parameters

The symbol N stood for "the maximum number of waiting tasks which can be allowed in the queue" in the definition of the queue. This was an example of a parameter. Parameters have many significances in elements. Generally, they represent variables upon which the behavior of the element depends, but which do not change with the status of the element. As far as this treatment is concerned, the parameters are constants. Other uses of parameters are as mean values of probability distributions, probabilities of branches, etc. Table I supplies the meanings of parameters, and the sets from which their values must be taken, for some symbols.

Because they are constants, parameters will be ignored in the remainder of this report. Any property of an element can be treated, in general, as a function of the parameters of the element without difficulty.

3. 2. 3 Epochs

Sometimes, when a symbol is being defined it is necessary to refer to points in time when something occurs (according to a probabilistic rule). Such a time is called an epoch.

To illustrate, the definition of a server having exponentially distributed holding times of mean $1/\mu$ will contain a statement to the effect "a service completion occurs with probability intensity μ whenever a task is present," and other statements to the effect "whenever a service completion occurs..." The times of "service completions" are epochs.

If epochs are points in time at which something occurs, then epoch classes are sets of those points selected by some semantic identification. If the probabilistic rule determining whether or not a particular time t is a member of an epoch class is purely determined within an element, then that epoch class will be termed an autogenous (self-generated) epoch class of the element, and its members are autogenous epochs. The semantics of autogenous epoch classes for the sample group of symbols are listed in a column of Table I. They are all characterized by the fact that they can be thought to occur spontaneously within the element, and are not "triggered" from actions (or epochs) of the associates.

Such triggering epoch classes, which will be called exogenous epoch classes, will be discussed in the section 3.2.4. Epochs and epoch classes will be defined more abstractly, in stochastic terms, in Section 6.

3.2.4 Port Variables

A rereading of the definition of the queue will recall a whole group of special phrases which refer to the influence of the associates of the queue. These phrases, like "the number of tasks acceptable to . . .," "the number of tasks available from . . .," "the number of tasks requested by . . ."

TABLE I

SEMANTICS OF STATE VARIABLES, PARAMETERS, AND AUTO-EPOCHS

Symbol	Name of Symbol	Parameter	State Variables	State Set	Auto-Epochs
	Queue	"The maximum allowed number of waiting tasks." (integer) : N	"The number of waiting tasks."	$\{0, 1, \dots, N\}$	None
	Exponential Server	"The mean rate of service for the occupied server." (real): γ	"The number of tasks in service."	$\{0, 1\}$	"Service Completions"
	Erlang Server	"The mean rate of service per phase." (real) : γ "The number of phases." (integer) : N	"The phase of current task."	$\{0, 1, \dots, N\}$	"Phase completions"
	Multiple Server	"The mean rate of service per task." (real) : γ "The number of servers." (integer) : N	"The number of tasks in service."	$\{0, 1, \dots, N\}$	"Service Completions"
	Poisson Arrival Element	"The mean rate of arrival." (real) : γ	None	$\{0\}$	"Arrivals"
	Exit	None	None	$\{0\}$	None
	Merge	None	None	$\{0\}$	None
	Overflow	None	None	$\{0\}$	None
	Balking Queue	"The mean rate of balk per waiting task." (real) : γ "The maximum allowed number of waiting tasks." (integer) : N	"The number of waiting tasks."	$\{0, 1, \dots, N\}$	"Balks"
	Random Distributor	"The a priori probability of upward selection." (real): p	None	$\{0\}$	None
	Random Collector	"The a priori probability of upward selection." (real): p	None	$\{0\}$	None
	Blocker	"The number of sensed tasks to produce blocking." (integer): N	None	$\{0\}$	None
	Contents Adder	None	None	$\{0\}$	None
	Random Switch	"The a priori probability of an upward setting." (real): p	None	$\{0\}$	None

or "the number of tasks offered by" represent variables sufficiently generalized that they continue to have meaning independently of the nature of the associate. Indeed, it will be seen shortly that they will have a technical meaning whenever the connection to the associate is syntactically allowable.

These phrases are used to describe what is "seen" through a connection either when looking from the queue toward the associate or from the associate toward the queue. The first two, "the number of tasks acceptable to ..." and "the number of tasks available from...", appear to play a role similar to that of the state variable when "..." is replaced by "the associate." Like state variables, they describe an observed condition of the element; but in this case the condition is external to the element. Such a variable will be called an exocondition (short for exogenous condition: a condition having external origin) at the designated port.

An exocondition of an element e is determined by the nature of the associate (call it e') of e , as a function of properties or conditions observed in the associate. This function, as yet not identified, is a property of e' . Its value, for example, is "the number of tasks acceptable to e' ." To e it is external, to e' it is internal. We will call this function an endocondition (for endogenous condition: a condition proceeding from within) function of e' , and its values will be called endoconditions. Clearly, if e' is to have exoconditions, then conversely e must also have an endocondition function.

The fact that there is a single variable "seen through the port" which, in combination with those "seen through" the other ports, is all that needs

to be known about the status of the rest of the network, is a very interesting one. It is a property of networks which seems natural, but which is, in fact, an assumption to be made about queueing networks. This assumption will be made here. It implies that all influences between elements are explicitly shown by the connections between them, and that influences from remote elements are achieved only through influences by them of the closer elements. Networks having this property will be called explicit. It appears that explicitness of a network is a necessary condition for the existence of a context-independent definition of elements.

Earlier we referred to epoch classes whose probabilistic rule of occurrence was external to the element under study. An example of this is the time-point represented by the phrase "whenever tasks are offered by (the associate)" in the definition of a queue. Such an epoch class is associated with a port and is called an exogenous epoch class (or exo-epoch class) at the port. Obviously, the associate must define this epoch, which it does by an endogenous epoch class. In English descriptions, one sees "whenever (epoch) and if (condition) then tasks will be offered."

Variables representing "the number of tasks offered by (the associate)," indicate an identifier of the specific interpretation attached to the action resulting from the epoch and will be called exocontrols for the element. The exocontrol of the element is determined by an endocontrol for the associate, which is a value of an endocontrol function. The "controls" always classify what is to happen when the corresponding epoch occurs.

It should be clearly noted that the two alternate phraseologies for each of our types of variables are distinctly associated with input and output ports (i. e., the "syntactic type" of the port), respectively. This is made clear in Table II, which tabulates the semantics. It is also clear from Table II that the endocondition of an input port of an element has the same meaning as the exocondition of an output port of its associate, and vice versa; and that this similarly holds true for -controls and -epochs. This indicates that when inputs are connected to outputs, the corresponding variables always have the same meaning, and that the eight variables (four for each port) in the connection reduce to just four distinctions.

In the syntactic description of section 2.2, two additional types of connection points were described, known as contents sensing and value collecting points. These connection point types were used in Figure 2.1 as part of the mechanism for describing blocking. They conceivably have other uses as well. Since contents sensing ports can only be connected to value collecting ports, the semantics of their port variables are interrelated in the same manner as the input and output ports: endoconditions of an element meaning the same thing as the exoconditions of its associate, etc. The semantics of these two port types are summarized in Table III. The controls were omitted because only one action can result from the exo-epoch. Thus the endo-control and exocontrol have no role and can be considered to be always zero. Similarly the cases indicated by a blank in Table III have no role to play in the sample vocabulary of the pictorial language, and are zero. Of course,

TABLE II
SEMANTICS OF INPUT AND OUTPUT PORT VARIABLES

Term and Symbol	Port Type	
	Input Port	Output Port
Endocondition v	"The number of tasks acceptable to (the element). "	"The number of tasks available from (the element). "
Exocondition x	"The number of tasks available from (the associate). "	"The number of tasks acceptable to (the associate). "
Endocontrol w	"The number of tasks requested by (the element). "	"The number of tasks offered by (the element). "
Exocontrol y	"The number of tasks offered by (the associate). "	"The number of tasks requested by (the associate). "
Endo-epoch	"When tasks are requested by (the element). "	"When tasks are offered by (the element). "
Exo-epoch	"When tasks are offered by (the associate). "	"When tasks are requested by (the associate). "

TABLE III

SEMANTICS OF CONTENTS SENSING AND VALUE COLLECTING PORTS

Term and Symbol	Port Type	Contents Sensing Port	Value Collecting Port
Endocondition v		"The number of tasks represented by (the element)."	
Exocondition x			"The number of tasks represented by (the associate)."
Endo-epoch		"When the number of tasks represented by (the element) changes."	
Exo-epoch			"When the number of tasks represented by (the associate) changes."

an extension of the pictorial vocabulary to other symbols might display a necessity for these variables to play a role.

The exocondition, endocondition, exocontrol, and endocontrol at a port will be known as port variables. If we designate them by x , v , y , and w respectively, their roles are suggested schematically by the arrows in Figure 3.2, where their port is represented as a short horizontal line, and the element as a box. This representation will be a handy aid in later discussion.

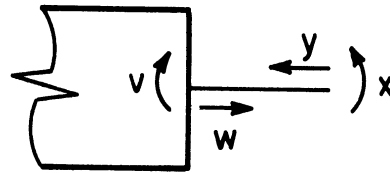


Figure 3.2. Schematic of port variables

In general, there will be no mathematical need to associate meanings like "number of tasks..." with these variables in order to define what we mean by the word "element." We assume only that one always connects ports together whose variables have mutually compatible meanings. Nevertheless, in illustrating an element—writing out its detailed characteristics—the semantics are helpful to the person who writes them out. Also, in determining whether a certain connection should be allowed (i. e., in syntax checking of the picture language), the consistency of these semantics is all-important. But in combining elements, forming new ones, and in writing out their formal mathematical descriptions the semantics are irrelevant.

Even the question of whether a port is an input or output port can be ignored. The ability to write algebraic descriptions of elements and to combine them is the objective of this report, so that these semantic descriptions of the variables will assume a low importance. They will be used only to illustrate, and sometimes to clarify.

3.2.5 Events

The verbal definition of the queue was expressed in two paragraphs. The first paragraph described the action which resulted from an exogenous epoch at the input port, while the second paragraph described the action which resulted from an exogenous epoch at the output port. The totality of action in an element known to occur when an exogenous epoch occurs will be called an exogenous event. Similarly, the action in an element known to occur when an autogenous epoch occurs will be called an autogenous event.

3.3 A DESCRIPTION OF SOME USEFUL ELEMENT TYPES

It will be useful now to define, verbally, several more of the element types partially described in Table I. This will clarify some of the fine points of behavior which might be confusing when these elements are used as illustrations of the algebraic network model.

3.3.1 The Exponential Server

A service completion occurs with probability intensity μ (the parameter; see Table I) whenever the number of tasks in service is

unity. When this occurs (an autogenous epoch), and the number of tasks acceptable to the associate at the output is zero, the task simply resumes service and no further action results from this epoch until it completes again. If the number of tasks acceptable to the output associate is non-zero, one task is offered to it. If, in this case, a task is available from the input associate one task is requested and the state remains unchanged (it was at unity). If a task is not available, the state is set to zero.

Whenever tasks are offered at the input and the state is unity, the number of tasks acceptable to the server is zero and no further action results. If the state is zero, one task is acceptable to the server, and the state is increased to unity.

When tasks are requested by the output associate, no tasks are available from the server, and no action results. The state remains unchanged.

3.3.2 The Poisson Arrival Element

An arrival occurs with probability intensity λ . When this occurs, one task is offered by the arrival element at the only port (which is an output). The state is always the same. When tasks are requested by the associate, no tasks are available from the element, and no action results.

3.3.3 The Random Distributor

When tasks are offered by the input associate, they are offered

at the outputs in a proportion determined as follows: A two-dimensional random walk experiment is conducted, with the probability of moving upward being p , and the probability of moving to the right being $(1-p)$. Whenever the vertical coordinate reaches the number of tasks acceptable to the upper output associate, or the horizontal coordinate reaches the number of tasks acceptable to the lower output associate, or the sum of both coordinates reaches the number of tasks offered by the input associate, the experiment stops. The coordinates indicate the number of tasks to be offered to the output associates. If the process is resumed until one of the first two conditions are met, then the sum of the coordinates indicates the number of tasks acceptable to the element at the input.

A similar strategy determines the result of a request for tasks by one of the output associates.

3.3.4 The Exit

When tasks are offered at the input port, the number of tasks acceptable by the element is infinite. No change in state occurs.

4. DEFINING A NETWORK MODEL

Since diagrams represent networks and symbols represent elements, and a diagram contains a collection of symbols, it is reasonable to suppose a network contains a set of elements. The discussion of semantics in the previous section has shown (or, at least, postulated) that it is meaningful to refer to elements in the abstract, without concern for their context. For this reason, one can indeed speak of a set \mathcal{E} of elements.

Furthermore, if we denote by \mathcal{P} the union of the ports of all the elements of \mathcal{E} , then a connection can be thought to consist of a pair of distinct elements of \mathcal{P} . The connection syntax in the pictorial language provided that each connection point would have at most one connecting line. Thus, the set C of connections can be termed a function mapping \mathcal{P} onto \mathcal{P} . Moreover, since " p_1 is connected to p_2 " implies also that " p_2 is connected to p_1 ," the connection function C is obviously its own inverse.

We shall therefore postulate that a Markovian network N consists of a set \mathcal{E} of elements, and a self-inverse function C , called a connection function. Or,

$$N = (\mathcal{E}, C). \quad (4.1)$$

Naturally, the task now is to find a suitable definition for an element e in \mathcal{E} . It will be seen in the next section that an element consists of a set of states, a set of ports, a set of autogenous events, and a set of exogenous events. The form which these latter two sets must take for the meaning conveyed to be sufficiently general will also be described.

Section 6 will show that the meaning conveyed is the correct one by firmly relating this model to the Kolmogorov equations of the underlying Markov chain.

5. ELEMENT REPRESENTATION

In this section we heuristically construct a suitable algebraic definition for an element. By successive examination of examples of elements, and of what we mean verbally by them, we will build up our model into a fairly complicated mathematical object. This constructive approach is taken in order to concretely explain the need for each feature of the model which is included. The true justification of the model as a consistent mathematical object will be treated in Section 6 where the definition will be related to the underlying Markov chain, and in Section 7 where the "connection" operation is defined. Thus, as far as mathematical development is concerned all but the last subsection of this section could be omitted. It is offered to improve credibility of the model, and to support the necessity of its complexity. It also illustrates many of the definitions in more practical terms.

Throughout this discussion, the semantics of Tables I and II will be used for the port-variables and the epoch classes.

5.1 STATES AND PORTS

Associated with an element e of a queueing network are two sets: a set of "states" S and a set of ports P . The set of states is the set of possible values of the state variables of the element. Thus each member of the set of states defines a possible condition of the element defined. Each port identifies a possible source of influence upon the element by

another element. A port p of the network is in the port set P of exactly one element $e \in \mathcal{E}$. Recall that a port was described as a point to which exactly one "connection" can be made.

The next order of business is to substitute a more precise definition for the relatively vague definition given for autogenous, endogenous, and exogenous events.

5.2 AUTOGENOUS EVENTS

An autogenous event for an element e has been defined in Section 3.2 as the action of an element which occurs when an autogenous epoch is known to have occurred. Let Λ_e be the set of autogenous epoch classes for element e , which may be empty. Then for each $\lambda \in \Lambda_e$, there is an autogenous event ξ^λ , and for the element e there is a set of these

$$\Xi_e = \{\xi^\lambda : \lambda \in \Lambda_e\} \quad (5.1)$$

which is empty if Λ_e is empty.

The purpose of the autogenous event is to describe the kind of information given in the verbal description of the action in the server when a service completion occurs or in a Poisson arrival element when an arrival occurs. This includes a description of the endogenous epoch classes generated, the endocontrols and endoconditions which play a role, the change in the local state which occurs, and the probability law which governs the epoch. These will in some way be dependent upon the identity of the epoch, the current local state, and the current exoconditions

and exocontrols.

All this will become more evident with examples. Consider first the element which presents the autogenous event in its purest form, the element used to describe a Poisson arrival process having mean rate of arrival γ .

5.2.1 The Poisson Arrival Element

This element was used in Figure 3.1 at the far left of the diagram, and was verbally described in Section 3.3. The graphic symbol used to describe it in network diagrams is shown in Figure 5.1. It has a single port (which is an output port) and a single local state; so that we might write

$$\begin{aligned} S &= \{0\} \\ P &= \{p\}. \end{aligned} \tag{5.2}$$

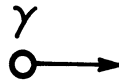


Figure 5.1

Furthermore, there is a single class λ of autogenous epochs, corresponding to the times when an "arrival" occurs. These epochs occur in any time interval of duration Δt with a probability $\gamma \Delta t$, when Δt is sufficiently small. (This is a property of the Poisson process.) Thus one says that epochs in λ occur with a probability intensity $\dot{\gamma}$. Each epoch produces the effect of generating an endogenous epoch class

(indicating an "offering of tasks") at the port p , with an endocontrol (indicating "the number of tasks offered") which is unity. There is, obviously, no change in local state possible.

We can note this description symbolically by saying that the autogenous event ξ^λ is a quadruple

$$\xi^\lambda = (p, 1, 0, \gamma) \quad (2.3)$$

indicating that the epoch causes an endogenous event at port p , with an endocontrol 1, a final state 0, and it occurs with probability intensity γ .

The picture presented here, however, is too simple. A more realistic idea of the form that an autogenous event should take can be seen by examining the exponential server. (For now, we are only concerned with its autogenous events.)

5.2.2 The Autogenous Events of the Exponential Server

It is evident from the symbol for this element (Figure 2.2) that this element has two ports, one input and one output. Let the state 0 represent the idle state, and 1 the busy state. Thus

$$\begin{aligned} S &= \{0, 1\} \\ P &= \{p_1, p_2\}. \end{aligned} \quad (5.4)$$

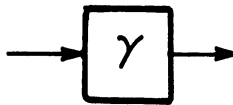


Figure 5.2

The only autogenous epochs are the "task completions," which have a probability intensity γ when the state is 1 and probability intensity 0 when the state is 0. This illustrates that it is necessary to regard the probability intensity of an event as a function of state. Further, each task completion can generate endogenous epochs at both ports p_1 and p_2 , since it may cause either an offering of a task at its output, a request for a task at its input, or both.

The endocontrol at these ports and the state resulting when this epoch occurs will, in general, depend upon the exoconditions observed at that time. If "the number of tasks acceptable to" the associate at p_2 is at least one, then it is known that the completed task can be sent, and therefore that a task can be requested from the associate at p_1 . In other words if the exocondition x_2 is positive ($x_2 > 0$), then the endocontrol w_1 ("the number of tasks requested") is unity. Otherwise it is zero. However, regardless of "the number of tasks acceptable to" the associate at port p_2 , a task can be "offered" through p_2 . That is, $w_2 = 1$. Clearly, the endocontrols (w_1, w_2) must be treated as functions of the exoconditions (x_1, x_2). A little consideration of the above discussion should bring the conclusion that the resulting state must also depend upon the exoconditions (x_1, x_2).

Analogously to Eq. (5.3), the above discussion could be described symbolically by saying that the autogenous event ξ^λ of this element is a quadruple

$$\xi^\lambda = (\{p_1, p_2\}, \beta, g, \mu) \quad (5.5)$$

where β is a vector valued function of (x_1, x_2) whose values are the values of the double $(w_1, w_2) = (\beta_1(x_1, x_2), \beta_2(x_1, x_2))$

$$\beta_1(x_1, x_2) = \begin{cases} 0 & \text{when } x_2 = 0 \\ 1 & \text{when } x_2 > 0 \end{cases} \quad (5.6)$$

$$\beta_2(x_1, x_2) = 1$$

where g is also a function of (x_1, x_2) :

$$g(x_1, x_2) = \begin{cases} 0 & \text{when } x_1 = 0 \text{ and } x_2 > 0 \\ 1 & \text{otherwise} \end{cases} \quad (5.7)$$

and where μ is a function of state $s \in S$:

$$\mu(s) = \begin{cases} 0 & \text{when } s = 0 \\ \gamma & \text{when } s = 1. \end{cases} \quad (5.8)$$

This quadruple indicates that the autogenous epoch causes endogenous epochs at p_1 and p_2 , with endocondition given by β_1 and β_2 , a target state given by g , and that it occurs with probability intensity given by μ .

5.2.3 A General Definition

The similarity between Eqs. (5.3) and (5.5) was, of course, intentional. Their mathematical form is identical if, in (5.3), we replace p by $\{p\}$, 1 by the unity function, 0 by the zero function, and γ by a constant function (all of appropriate domain). The two examples are sufficient

to motivate the general (abstract) definition of an autogenous event given below. It should be noted, however, that neither example had more than one autogenous event, even though such a situation is possible. Thus, in addition to the sets S and P , a set Ξ of autogenous events must take a place in the definition of an element.

The definition below also takes cognizance of some other possibilities in the forms of the functions which were not demonstrated in the examples. For example, it is quite certainly true that endocontrols can take on values greater than unity. How else would such things as bulk arrival or service be described? The philosophy followed is to keep whatever usable generality appears inherent in the examples and notation.

In general, every autogenous event ξ^λ can be defined in a quite natural way as a function of $s \in S$, whose values consist of quadruples

$$\xi_s^\lambda = (Q^\lambda, \beta_s^\lambda, g_s^\lambda, \mu_s^\lambda) \quad (5.9)$$

where the components are defined as follows:

1. The endogenous epoch set Q^λ is a set of ports, $Q^\lambda \subseteq P$ at which endogenous events are identified.
2. Let $\mathcal{J}^\lambda = I_{p_1} \times I_{p_2} \times \dots \times I_{p_n}$, where $\{p_1, p_2, \dots, p_n\} \triangleq Q^\lambda$, and where $I_p = \{0, 1, 2, \dots\}$ for all $p \in P$. Then the endocontrol function β_s^λ is a function on \mathcal{J}^λ to \mathcal{J}^λ , for each $s \in S$. If we write this function out as

$$(w_{p_1}, w_{p_2}, \dots, w_{p_n}) = \beta_s(x_{p_1}, x_{p_2}, \dots, x_{p_n}) \quad (5.10)$$

then w_p represents the endocontrol at port p , and x_p represents the exocondition at port p , for all $p \in Q^\lambda$.

3. The target state function g_s^λ is a function on \mathcal{J}^λ to S , for each $s \in S$. It represents the target state as a function of the original state and the exoconditions at the ports in Q^λ . In other words $g_s(x_{p_1}, x_{p_2}, \dots, x_{p_n})$ is the value of the target state when the element is originally in state s and sees exoconditions x_{p_1}, \dots, x_{p_n} at ports p_1, \dots, p_n respectively.

4. The probability intensity μ_s^λ is a non-negative real number. It represents the probability intensity of the occurrence of the epoch generating the autogenous event ξ^λ , as a function of the original state.

It remains to be shown, in Section 6, that this definition is unambiguous. It also remains to be shown that it is sufficiently general to suit all potential needs within specified assumptions. The latter, unfortunately, appears to be only provable by experience. Nevertheless, the question of generality will also be explored in Section 6.

5.3 EXOGENOUS EVENTS

An exogenous event for an element e is that object which identifies the effects upon e observed when an exogenous epoch occurs at one of its ports. A more detailed and precise definition for exogenous events is now sought.

The epochs which determine such events are defined by autogenous

events in other elements in a network. They are observed in the element e only by the occurrence of an endogenous epoch class in an associate of e .

The appearance of an exogenous event at a port of e obligates e to define an endocondition at that port which serves as an exocondition to the adjacent element. It also requires the specification of the observed effect of the epoch upon e . The effect may be either the generation of endogenous epoch classes at other ports, or a change in state, or both. A further examination of the exponential server element will help clarify these representational needs.

5.3.1 The Exogenous Events of the Exponential Server

We observe that an exogenous epoch for this element is the "offering of tasks" into port p_1 . The first necessity is to define the endocondition v_1 , the "number of tasks which can be accepted by e ," as it is observed at one of these epochs:

$$v_1 = \alpha_1(s) = 1-s, \quad \text{all } s \in S \quad (5.11)$$

where $S = \{0, 1\}$ as before. This indicates that a task can be accepted only if the server is idle. We further observe that no endogenous epoch classes are generated by this epoch, and that the state will increase from 0 to 1 whenever the exocontrol y_1 ("the number of tasks offered by the associate") at port p_1 is not zero.

This description can be symbolically noted by saying that this exogenous event ξ_1 is a triple

$$\xi_1 = (\{p_1\}, \alpha_1, g_1), \quad (5.12)$$

signifying that the epoch class is determined from port p_1 , that the endocondition at p_1 is given by the function α_1 (Eq. (2.11)), and that the target state after the epoch occurs will be given by the function g_1

$$g_1(s, y_1) = \begin{cases} 0 & y_1 = 0 \text{ and } s = 0 \\ 1 & y_1 > 0 \text{ or } s = 1. \end{cases} \quad (5.13)$$

We can also observe that a second exogenous epoch class for this element is the "requesting of a task" by the associate at port p_2 . However, since a server can only transmit tasks at its output during "job completion" epochs, the consequences of the request are simple to describe: an endocondition of 0 is always shown and there are no endogenous epoch classes or changes in state. This exogenous event ξ_2 could be described by the triple

$$\xi_2 = (\{p_2\}, 0, g_2) \quad (5.14)$$

where

$$g_2(s) = s$$

using the same notation as for Eq. (5.12).

The description used in Eqs. (5.12) and (5.14) does not take account of the possibility that an exogenous epoch class may generate an endogenous epoch class at a port other than the one defining the epoch. This possibility gives rise to more complexity in the description of an

exogenous event, and is well illustrated in the attempt to describe the behavior of the "queue" element.

5.3.2 The Queue

This element, whose symbol appears in Figure 5.3 must accept inputs of groups of tasks and hold them until they are requested (in groups) from other elements. It has two ports, an input (p_1) and an output (p_2). Its state is described by the "number of waiting tasks present," and is bounded by a constant, N . The queue is entirely controlled from outside, having no autogenous epoch classes and thus no autogenous events. Thus,

$$\begin{aligned} P &= \{p_1, p_2\} \\ S &= \{0, 1, 2, \dots, N\} \\ \Xi &= \phi \end{aligned} \tag{2.15}$$



Figure 5.3

The exogenous epoch at port p_1 is the "offering of tasks by the associate." If an element connected at p_2 "can accept" any of these tasks, they will be passed on. Thus, the exogenous epoch at p_1 causes an endogenous event to be generated at p_2 . The endocondition v_1 ("the number of tasks acceptable") at p_1 , and the target state g_1 ("the number of waiting tasks"), will depend upon the value of the exocondition x_2 ("the number of

tasks acceptable'' at p_2 and the exocontrol y_1 (''the number of tasks offered'' at p_1). In fact, then, we can write

$$v_1 = \alpha_1(s, x_2) = N - s + x_2 \quad (5.16)$$

signifying that the number of tasks acceptable at p_1 is the capacity of the queue, less the number already present, plus the number which can be passed on through p_2 . We can also write

$$g_1(s, x_2, y_1) = \min(N, \max(0, s + y_1 - x_2)) \quad (5.17)$$

signifying that the number in queue after the epoch occurs is the number present before the epoch, plus the number offered by the associate at the input, less the number acceptable by the associate at the output (but at least zero and at most N). The endocontrol w_2 (''the number of tasks offered'' associated with the endogenous epoch class at p_2 can be written

$$w_2 = \beta_2(s, y_1) = s + y_1 \quad (5.18)$$

signifying that the number offered to the output associate is the number in queue plus the number offered by the input associate.

This exogenous event is considerably more complicated than the first (i. e., Eq. (5.12)), but it can be described by the quintuple

$$\xi_1 = (\{p_1\}, \{p_2\}, \alpha_1, \beta_2, g_1) \quad (5.19)$$

signifying that the epoch is determined from port p_1 , that this epoch generates an endogenous epoch class at p_2 , with an endocontrol given by the value of the function β_2 , that the endocondition at p_1 is given by the value

of the function α , and that the state immediately after this epoch is given by the value of the function g_1 . The triple used in the previous example can be expanded to a special case of this more general form.

The second exogenous event of the queue describes the effect of an epoch determined at port p_2 , representing "a request for tasks by the associate at p_2 ." The endocondition v_2 , endocontrol w_1 , and target state g_2 for this event are analogously determined by the equations

$$v_2 = \alpha_2(s, x_1) = s + x_1 \quad (5.20)$$

$$g_2(s, x_1, y_2) = \min(N \max(0, s - y_2 + x_1)) \quad (5.21)$$

$$w_1 = \beta_1(s, y_2) = N - s + y_2 \quad (5.22)$$

using an obvious notational convention for the subscripts. This event ζ_2 is described by the quintuple

$$\zeta_2 = (\{p_2\}, \{p_1\}, \alpha_2, \beta_1, g_2) \quad (5.23)$$

5.3.3 A General Form (Deterministic)

In the examples, there has been at most a single port identifying the source of an exogenous epoch. If we assume that this will always be the case, then the set of exogenous events are in one-to-one correspondence with the set of ports. This suggests that a more efficient format for the description of exogenous events would be as a quadruple which is a function defined on the set of ports P , rather than as the quintuples used in Eqs. (5.19) and (5.23).

If we adopt this convention, then there is a single exogenous event function ζ^s defined on P . Attempting to provide enough generality in the form which this function takes (general in the same sense as our previous definition of the autogenous event function), this exogenous event function would have to take the form of a quadruple

$$\zeta_s^p = (Q^p, \alpha_s^p, \beta_s^p, g_s^p), \quad p \in P, \quad s \in S \quad (5.24)$$

whose components are defined as follows:

1. The endogenous epoch set Q^p is a set of ports, $Q^p \subseteq P$ at which endogenous events are generated by ζ_s^p .

2. Define an integer space \mathcal{J}^p by $\mathcal{J}^p = I_{p_1} \times I_{p_2} \times \dots \times I_{p_n}$ where $\{p_1, \dots, p_n\} \triangleq Q^p$, and where $I_q = \{0, 1, 2, \dots\}$ for all $q \in P$ as before. Then the endocondition function α_s^p is a function on \mathcal{J}^p to I_p for each $p \in P, s \in S$. If we write this function out as

$$v_p = \alpha_s^p(x_{p_1}, x_{p_2}, \dots, x_{p_n}) \quad (5.25)$$

then v_p represents the endocondition at port p , and $x_q, q \in Q^p$ represents the exocondition at port q .

3. The endocontrol function β^p is a function on $\mathcal{J}^p \times I_p$ to \mathcal{J}^p for each $p \in P$. Writing the function out as

$$(w_{p_1}, w_{p_2}, \dots, w_{p_n}) = \beta_s^p(x_{p_1}, x_{p_2}, \dots, x_{p_n}, y_p) \quad (5.26)$$

then $w_q, q \in Q^p$, represents the endocontrol at port q when the element is in state s and "excited" by an exogenous event at port p , y_p represents the

exocontrol at port p , and x_γ , $\gamma \in Q^p$, represents the exocondition at port γ .

4. The target state function g_s^p is a function on $\mathcal{Q}^p \times I_p$ to S , for each $s \in S$. Thus $g_s^p(x_{p_1}, x_{p_2}, \dots, x_{p_n}, y_p)$ is the value of the target state when: (a) the element is in state s , (b) an exogenous event arrives at port p , and (c) the element sees the exoconditions $x_{p_1}, x_{p_2}, \dots, x_{p_n}$ at ports p_1, \dots, p_n , respectively (where $\{p_1, \dots, p_n\} = Q^p$ as before), and y_p is the exocontrol at port p .

This rather formidable object ζ_s^p is, as we have seen, a natural result of attempting to find a description which fits the exogenous events of the foregoing examples in some abstraction. It is evidently complex because our intuitive concept of an exogenous event is surprisingly complex. Worse yet, as will be seen in the next section, even this is not quite general enough.

5.3.4 Random Effects—The Random Switch

If a state and exogenous epoch is given, then the endoconditions returned, the endogenous epoch classes generated, the endocontrols, and the target state may not all be deterministic. Rather, these variables can be random variables not totally determined by the particular exogenous epoch class which otherwise identifies them. This requires a further generalization of the mathematical model for an exogenous event.

A very simple example of this problem can be found in the random switch symbolized in Figure 5.4. The operation of this switch is as follows:

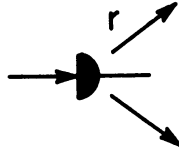


Figure 5.4. The random switch.

Each time a task or tasks are offered at its input, or requested from an output, the switch is set. With probability r it is set between the input and the upper output, and with probability $1-r$ it is set between the input and the lower output. After the switch is set the tasks flow, to the extent possible, from the input to the selected output.

This element has three ports, a single state, and no autogenous events:

$$\begin{aligned} P &= \{p_0, p_1, p_2\} \\ S &= \{0\} \\ \Xi &= \phi. \end{aligned} \tag{5.27}$$

An exogenous epoch determined at port p_0 (the input) has two possible effects. With probability r it produces an endogenous epoch class at p_1 , returns an endocondition which is equal to the exocondition at p_1 , sends an endocontrol equal to the exocontrol at p_0 , and remains at state 0. In the notation of Eq. (5.24)

$$\zeta_{s}^{p_0}(s) = (Q_{s,\alpha}^{p_0}, \alpha_{s,\beta}^{p_0}, \beta_{s,g}^{p_0}, g_{s}^{p_0}) \tag{5.28}$$

where

$$\begin{aligned}
Q^{p_0} &= \{p_1\} \\
v_0 &= \alpha_s^{p_0}(x_1) = x_1 \\
w_1 &= \beta_s^{p_0}(x_1, y_0) = y_0 \\
&= g_s^{p_0}(x_1, y_0) = 0.
\end{aligned} \tag{5.29}$$

With probability $1-r$, on the other hand, the roles of p_1 and p_2 are reversed so the exoevent is

$$\xi_s^{p_0} = (Q^{p_0}, \alpha_s^{p_0}, \beta_s^{p_0}, g_s^{p_0}) \tag{5.30}$$

where

$$\begin{aligned}
Q^{p_0} &= \{p_2\} \\
v_0 &= \alpha_s^{p_0}(x_2) \wedge x_2 \\
w_2 &= \beta_s^{p_0}(x_2, y_0) = y_0 \\
g_s^{p_0}(x_2, y_0) &= 0.
\end{aligned} \tag{5.31}$$

A handy notation would treat the exo-event set Z for the element as a set valued function on P whose value consists of exo-events over an index set which is distinctive for each port. Thus

$$Z(p) = \{\xi^\sigma : \sigma \in \Sigma_p\}, \quad p \in P \tag{5.32}$$

Then the "exoevent" $\xi_s^{p_0}$ corresponding to Eq. (5.28) would be coupled with the probability r , and written

$$\zeta_{\sigma_1}^{\sigma_1} = (Q^{\sigma_1}, \alpha^{\sigma_1}_{\sigma_1}, \beta^{\sigma_1}_{\sigma_1}, g^{\sigma_1}_{\sigma_1}, r) \quad (5.33)$$

where Σ_{p_0} contains σ_1 . The "exoevent" $\zeta^{\sigma_1}_{p_0}(s)$ becomes

$$\zeta_{\sigma_2}^{\sigma_2} = (Q^{\sigma_2}, \alpha^{\sigma_2}_{\sigma_2}, \beta^{\sigma_2}_{\sigma_2}, g^{\sigma_2}_{\sigma_2}, (1-r)) \quad (5.34)$$

and we can write $\Sigma_{p_0} = \{\sigma_1, \sigma_2\}$.

The exo-events for the other ports are similarly derived, as

$$\zeta^{\sigma_3}(s) = (\{p_0\}, \alpha'_1, \beta'_1, 0, r) \quad (5.35)$$

$$\zeta^{\sigma_4}(s) = (\emptyset, 0, 0, 0, (1-r)) \quad (5.36)$$

where $\Sigma_{p_1} = \{\sigma_3, \sigma_4\}$, 0 represents the zero function, and where

$$\alpha'_1(x_0) = x_0,$$

and

$$\beta'_1(x_0, y_1) = y_1.$$

For port p_2 the roles of p_1 and p_2 above are interchanged.

5.3.5 A General Form (Stochastic)

As implied by the foregoing example, the exogenous events of an element can be described by a set $Z(p) = \{\zeta^{\sigma} : \sigma \in \Sigma_p\}$, $p \in P$ with each event ζ^{σ} being a function on S of the form

$$\zeta^{\sigma}_s = (Q^{\sigma}, \alpha^{\sigma}_s, \beta^{\sigma}_s, g^{\sigma}_s, \pi^{\sigma}_s), s \in S \quad (5.37)$$

where $Q^{\sigma}, \alpha^{\sigma}_s, \beta^{\sigma}_s$, and g^{σ}_s are defined exactly as they were in Section 5.3.3, except that p is now defined as the port p such that $\sigma \in \Sigma_p$. The

symbol π_s^σ represents a function whose range is in the interval $[0, 1]$. Its domain may be quite complex and will be treated in full after some additional theory is presented in Section 6. A set-valued function Z , mapping a set of ports onto a set of exogenous events, will be called an exo-event set.

5.4 FULL DEFINITION OF AN ELEMENT

With the description of the state set S , the port set P , auto-event set Ξ , and the exo-event set Z for an element e , we appear to have described all of the technical meaning represented by the element. We now turn this around and define the element e as the collection of four such sets as defined. In other words, we let

$$e = (S, P, \Xi, Z) \quad (5.38)$$

This completes our definition of element and network, except for the tricky considerations regarding π_s^σ in the definition of the exoevent, which will be clarified later.

To summarize,

- (1) S is a set of objects called states
- (2) P is a set of objects called ports
- (3) Ξ is a set of objects called autogenous events
- (4) Z is a set valued function on P to sets of exogenous events
- (5) An autogenous event ξ is a quadruple, a function on S .

$$\xi_s = (Q, \beta_s, g_s, \mu_s), \quad s \in S$$

where

- (a) Q is a set of ports, called an endo-epoch set.
 - (b) β_s , for all $s \in S$, a function of as many integer variables as there are ports in Q (representing the endoconditions at these ports). Its range has an equal number of (integer) components (representing the endocontrols at the ports).
 - (c) g_s , for all $s \in S$, is a function of as many integer variables as there are ports in Q (also representing the endoconditions), into the set S . It represents the target state of the event.
 - (d) μ_s , for all $s \in S$, is a positive real number (representing the probability intensity occurrence of the event).
- (6) An exogenous event ζ is a quintuple, a function on S

$$\zeta_s = (Q, \alpha_s, \beta_s, g_s, \pi_s), \quad s \in S$$

where

- (a) Q is a set of ports, called an endo-epoch.
- (b) α_s , for all $s \in S$, is a function of as many integer variables as there are ports in Q (representing exoconditions at these ports), into the set of nonnegative integers (representing the endocondition at the port p for which $\zeta_s \in Z(p)$).
- (c) β_s , for all $s \in S$, is a function of as many integer variables as there are ports in Q , plus one (representing the endoconditions and an exocontrol at p defined above.) Its range

has as many (integer) components as there are ports in Q (representing the endocontrols at these ports).

- (d) g_s , for all $s \in S$, is a function of the same variables as β_s (above) into S . It represents the target state of the event.
- (e) π_s , for all $s \in S$, is a function of the same variables as β_s (above), as we shall see, and is onto the interval $[0, 1]$.

It represents a probability.

Table III formally lists the defining sets and functions for some of the more common elements. It should be, for the most part, self-explanatory, except for a few notational conventions. The parameter symbols (γ, r, N) from Table I are used without further explanation. The subscript "1" on a port is always for an input port, while a "2" is for an output port. More than one port of a type are distinguished by primes. The exoconditions and exocontrols use the main symbols x and y as usual, with the subscripts and primes borrowed from their respective ports. Endocontrols, as vectors, are ordered in the same sequence as the elements of the corresponding endoevent set. An asterisk (*) is used where a function is degenerate, having null range and domain, as when an endocontrol vector has no components. * For a set of numbers A , and any variable a , the symbol $I_a(A)$ represents the function which has unity value whenever a is in A , and is

* An alternate notation is to use the symbol \emptyset to denote the function having null range and null domain.

TABLE IV
ELEMENT DEFINITIONS¹

Element Type	State Set S	Port Set P	Autogenous Events $\xi_S^\lambda \in \Xi$				q	Exogenous Events ζ_S^σ				
			Endoevent Set Q^λ	Endocontrol $\beta_S^\lambda(x)$	Target State $g_S^\lambda(x)$	Intensity $\mu(x)$		Endoevent Set Q^σ	Endocondition $\alpha_S^\sigma(x)$	Endocontrol $\beta_S^\sigma(x, y_q)$	Target State $g_S^\sigma(x, y_q)$	Probability π_S^σ
Queue	{0,1,...,N}	{p ₁ , p ₂ }	None				p ₁	{p ₂ }	N-s+x ₂	s+y ₁	$[s+y_1-x_2]_0^N$	1
							p ₂	{p ₁ }	s+x ₁	N-s+y ₂	$[s-y_2+x_1]_0^N$	1
Server	{0, 1}	{p ₁ , p ₂ }	$\begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix}$	$\begin{pmatrix} I_{x_2}(J^+), \\ 1 \end{pmatrix}$	$I_{x_1}(J^+)I_{x_2}\{0\}$	$\gamma I_S\{1\}$	p ₁	ϕ	1-s	*	$1-I_{y_1}(\{0\})I_S(\{0\})$	1
							p ₂	ϕ	0	*	s	1
Arrival	{0}	{p ₂ }	{p ₂ }	1	0	γ	p ₂	ϕ	0	*	0	1
Exit	{0}	{p ₁ }	None				p ₁	ϕ	∞	*	0	1
Random Switch	{0}	{p ₁ , p ₂ ', p ₂ '}	None				p ₁ '	{p ₂ '}	x ₂ '	y ₁	0	r
							p ₁ '	{p ₂ '}	x ₂ '	y ₁	0	(1-r)
							p ₂ '	{p ₁ '}	x ₁	y ₂ '	0	r
							p ₂ '	ϕ	0	*	0	(1-r)
	{0}	{p ₁ , p ₁ ', p ₂ '}	None				p ₂ '	ϕ	0	*	0	r
							p ₂ '	{p ₁ '}	x ₁	y ₂ '	0	(1-r)
							p ₁ '	{p ₂ '}	x ₂	y ₁ '	0	1
							p ₁ '	{p ₂ '}	x ₂	y ₁ '	0	1
Merge	{0}	{p ₁ , p ₁ ', p ₂ '}	None				p ₂	$\begin{Bmatrix} p_1' \\ p_1'' \end{Bmatrix}$	x ₁ ' + x ₁ '	$\begin{pmatrix} [y_2/2]_0^{x_1'} \\ y_2 - [y_2/2]_0^{x_1'} \end{pmatrix}$	u	.5
								$\begin{Bmatrix} p_1' \\ p_1'' \end{Bmatrix}$	x ₁ ' + x ₁ '	$\begin{pmatrix} y_2 - [y_2/2]_0^{x_1''} \\ [y_2/2]_0^{x_1''} \end{pmatrix}$	0	.5
								$\begin{Bmatrix} p_1' \\ p_1'' \end{Bmatrix}$	x ₁ ' + x ₁ '	$\begin{pmatrix} [y_2/2]_0^{x_1'} \\ y_2 - [y_2/2]_0^{x_1'} \end{pmatrix}$	u	.5
								$\begin{Bmatrix} p_1' \\ p_1'' \end{Bmatrix}$	x ₁ ' + x ₁ '	$\begin{pmatrix} y_2 - [y_2/2]_0^{x_1''} \\ [y_2/2]_0^{x_1''} \end{pmatrix}$	0	.5

¹For notation, see text.

zero otherwise. That is, $I(A)$ is the indicator function of the set A . The symbol J^+ is used to represent the set of positive integers $\{1, 2, \dots\}$, so that $I_a(J^+)$ has unity value whenever a is an integer greater than zero, and is zero otherwise. For any variables a, b, c , the symbol $[a]_b^c$ signifies the integer part of a not exceeding c or lying below b . In other words

$$[a]_b^c = \min\{b, \max\{c, \text{int}(a)\}\}. \quad (5.39)$$

6. THE NETWORK AS A MARKOV CHAIN

The pictorial language describes not only Markovian systems, but also Markov chains. At this point the network model will be related to the Markov chains, and will thereby be given a more rigorous foundation. So far, the chief justification for our constructs has been that they seem to parrot what one does in ordinary verbal description. This is, of course, not an adequate basis for constructing a network model.

In this section, new probabilistic definitions will be given for many of the concepts more loosely used in the foregoing development of the model. These new definitions, and the mathematical system they evolve, represent still another "translation;" this being the last one. Because they are more precise, these definitions are also somewhat more general. Nevertheless, the same terms and symbols will be used for the new as for the old. To have created a whole new symbology would have been still more confusing.

One major difference in the viewpoint of this section from the previous one is that the network is viewed as a whole, and then gradually broken down into its component parts, rather than the reverse. This is a consequence of the fact that Markov chains are not naturally broken up into things like elements and connections, and so our development must proceed from the well known theory toward a more conventional view of Markov chains. Thus, the starting point is the entire network.

6.1 THE KOLMOGOROV AND EQUILIBRIUM EQUATIONS

The only concern of this report is with continuous-time, finite Markov chains with stationary transition probabilities. This model consists of a one parameter family of random variables $\{Z_t, t \geq 0\}$ having the property that, for any real t_1, t_2, t_3 such that $0 \leq t_1 \leq t_2 < t_3$, and for any z_1, z_2, z_3 in the range of Z_t ,

$$\text{pr}\{Z_{t_3} = z_3 | Z_{t_2} = z_2, Z_{t_1} = z_1\} = \text{pr}\{Z_{t_3} = z_3 | Z_{t_2} = z_2\}, \quad (6.1)$$

the latter being a function of z_2, z_3 , and $(t_3 - t_2)$. The range of the random variable Z_t is called the state set, will be denoted by \mathcal{L} and will always be finite here.

It is a well-known (cf. theorem II.18.3 and corollary II.19.2 of Ref. 1) property of finite-state continuous-time Markov chains with stationary transition probabilities that the process is completely determined by the transition intensities, defined as

$$u_{ij} \triangleq \frac{d}{dt} \text{pr}\{Z_t = j | Z_0 = i\} \big|_{t=0} \quad (6.2)$$

for all $i, j \in \mathcal{L}$. The transient state probabilities for this process

$P_{ij}(t)$ where by definition $p_{ij}(t) = \text{pr}\{z_t = j | z_0 = i\}$, are found from the Kolmogorov differential equations

$$\frac{d}{dt} p_{ij}(t) = \sum_{k \in \mathcal{L}} u_{kj} p_{ik}(t). \quad (6.3)$$

The equilibrium state probabilities, $\pi_i, i \in \mathcal{L}$, are found from a solution

of the system of algebraic equations

$$\sum_{i \in \mathcal{A}} u_{ij} \pi_i = 0 \quad (6.4)$$

for all $j \in \mathcal{A}$. It is these latter equations, known as the equilibrium equations, which the Recursive Queue Analyzer² or similar algorithms are capable of efficiently solving.

If the state set \mathcal{A} consists of a sequence consecutive integers $(1, 2, \dots, n_s)$, then Eq. (6.4) takes the familiar matrix form of a homogeneous set of equations

$$\pi U = 0. \quad (6.5)$$

However, the state can, in general, be regarded as an abstract object and \mathcal{A} an abstract set.

6.2 A VECTOR VALUED MARKOV CHAIN

For networks as treated here, the state will be described by a vector consisting of a number of components (the state-variables of all the elements). Thus, if $Z_t(1)$ is a queue length in the network, and $Z_t(2)$ is the number of servers of a particular multiple server which are occupied, and $Z_t(3)$ is some other queue length, etc., then

$$Z_t = (Z_t^{(1)}, Z_t^{(2)}, \dots, Z_t^{(n)}, \quad t \geq 0 \quad (6.6)$$

is a vector-valued Markov chain, which will be denoted by Z . Letting S_k be the set of possible values of $Z_t(k)$, then $\mathcal{A} \subseteq S_1 \times S_2 \times \dots \times S_n$ is the state set for Z . The " \subseteq " symbol is used instead of the "=" symbol

because certain combinations of the components may not be possible or necessary. The components $Z_t^{(k)}$, treated as random processes, will be called process components, while Z_t , the vector, will be called simply the process. Each process component is a continuous parameter, discrete state random process, but in general only the process as a whole has the Markovian property.

The components $Z_t^{(k)}$ correspond to the variables we called state variables in the earlier sections, and are therefore identified with the elements. Each component will be associated with exactly one element, although an element may be associated with more than one component. In general, samples of these components are defined only as elements of their respective sets S_k , and need not be assumed to be integers. (The example has been given that a state component may designate the current pattern of paths established through an element representing a switching network, these patterns having no direct correspondence to numbers.) Nevertheless, each component state set S_k must be finite, so it will always be possible to put the states in correspondence to integers.

6.3 EPOCHS

The role played by epochs is a vital one to the treatment of Markov chains as networks.

6.3.1 A Definition

Each nonzero value of u_{ij} for $i \neq j$ signifies that at any time when Z_t is in state i (i. e., when Z_t has value i) there is a chance that Z_t may jump to state j . This chance is measured by the probability

$$\text{pr}\{Z_{t+\Delta t} = j | Z_t = i\} = u_{ij}\Delta t \quad (6.7)$$

for Δt sufficiently small and $i \neq j$. (The value of u_{ij} for $i=j$ is given by

$$1 - \text{pr}\{Z_{t+\Delta t} = i | Z_t = i\} = -u_{ii}\Delta t \quad (6.8)$$

for Δt sufficiently small. Since

$$\sum_j u_{ij} = 0 \quad \text{all } i \in \mathcal{S}, \quad (6.9)$$

we will not be concerned with u_{ii} —it is readily found when all other u_{ij} are known.) The value in time at which this jump occurs is an epoch of the process. Thus $u_{ij}\Delta t$ represents the probability, given that Z_t is in state i at time t , that an epoch which takes i to j occurs in the interval $[t, t+\Delta t]$.

This object represents an abstraction of the concept of epoch described in section 3. Let the sample space for the Markov chain Z_t be the set Ω . Then, for each $\omega \in \Omega$ there is a set $\tau(\omega)$ of epochs marking the times of all jumps in $Z_t(\omega)$. Thus τ is a set-valued random variable. When one refers to the object "completions of service in server 3" or "arrivals at the input to queue 1," as was done in section 3, one is referring to a subset of $\tau(\omega)$ for each $\omega \in \Omega$. An

individual "completion of service" refers to a member of this subset, which is also a member of $\tau(\omega)$, and hence is also an epoch.

6.3.2 Epoch Classes

Let an arbitrary subset of $\tau(\omega)$ be represented as $\nu(\omega)$ for all $\omega \in \Omega$. Then ν is a set valued random variable which is called an epoch class. This, also, represents an abstraction of the concept of epoch class described in section 3. Phrases like "completions of service..." identify epoch classes, in this more precise sense.

Epoch classes are not necessarily disjoint subsets of τ . For example, a "service completion of server 3 when switch 1 is in its left position" identifies an epoch class, but so do a "service completion of server 3" and "any occurrence when switch 1 is in its left position." The first is, in fact, the intersection of the second and third.

6.3.3 Partitions of the Epoch Set

Suppose, however, that the set τ of epochs of $\{Z_t\}$ is partitioned by a set $\mathcal{N} = \{\nu_1, \nu_2, \dots, \nu_m\}$ of (disjoint) epoch classes, where

$$\bigcup_{k=1}^m \nu_k(\omega) = \tau(\omega) \quad (6.10)$$

and

$$\nu_{k_1}(\omega) \cap \nu_{k_2}(\omega) = \phi \quad (6.11)$$

for all k_1 and k_2 in $\{1, 2, \dots, m\}$, and for all $\omega \in \Omega$. Such a partitioning allows Eqs. (6.3) and (6.4) to be rewritten in a very useful form.

Let φ be a function representing the probability intensity of occurrence of an arbitrary epoch class ν , given only that the process is in state i . Thus

$$\varphi(\nu, i) \triangleq \lim_{\Delta t \rightarrow 0} \frac{\text{pr}\{\nu \cap [0, \Delta t] \neq \phi \mid Z_0 = i\}}{\Delta t}, \quad (6.12)$$

for all epoch classes ν and all $i \in \mathcal{S}$. Recall that, because of the stationarity of $\{Z_t\}$, we are free to observe these probability intensities merely at $t = 0$. To avoid the unsightly expression " $\nu \cap [0, \Delta t] = \phi$," we shall henceforth (where ν is any epoch class) use the more graphic phrase " ν occurs in Δt " to mean the same thing. Thus

$$\varphi(\nu, i) \triangleq \lim_{\Delta t \rightarrow 0} \frac{\text{pr}\{\nu \text{ occurs in } \Delta t \mid Z_0 = i\}}{\Delta t}. \quad (6.12a)$$

for all epoch classes ν and all $i \in \mathcal{S}$. Furthermore, let h be a function representing the probability that the epoch, having just occurred, results in a transition to j . That is,

$$h(\nu, i, j) \triangleq \lim_{\Delta t \rightarrow 0} \text{pr}\{Z_{\Delta t} = j \mid \nu \text{ occurs in } \Delta t \text{ and } Z_0 = i\}, \quad (6.13)$$

for all i and j in \mathcal{S} . We note, from Eq. (6.2), that

$$u_{ij} \triangleq \frac{d}{dt} \text{pr}\{Z_t = j | Z_0 = i\} \big|_{t=0}.$$

Also,

$$\varphi(\nu, i)h(\nu, i, j) = \lim_{\Delta t \rightarrow 0} \frac{\text{pr}\{Z_{\Delta t} = j \text{ and } \nu \text{ occurs in } \Delta t | Z_0 = i\}}{\Delta t} \quad (6.14)$$

and,

$$\sum_{\nu \in \mathcal{N}} \varphi(\nu, i)h(\nu, i, j) = \lim_{\Delta t \rightarrow 0} \frac{\text{pr}\{Z_{\Delta t} = j \text{ and } \tau \text{ occurs in } \Delta t | Z_0 = i\}}{\Delta t} \quad (6.15)$$

since \mathcal{N} is a finite partitioning of τ . We observe that for $i \neq j$ it is impossible for $Z_{\Delta t}$ to be equal to j when $Z_0 = i$ if some epoch did not occur in the interval $[0, \Delta t]$. Consequently

$$\text{pr}\{Z_{\Delta t} = j \text{ and } \tau \text{ occurs } \Delta t | Z_0 = i\} = \text{pr}\{Z_{\Delta t} = j | Z_0 = i\} \quad (6.16)$$

for all $\Delta t \geq 0$ and all $i \neq j$. From this it is readily concluded that

$$\sum_{\nu \in \mathcal{N}} \varphi(\nu, i)h(\nu, i, j) = u_{ij} \quad (6.17)$$

for all $i, j \in \mathcal{L}$, $i \neq j$.

What this means is that, for an arbitrary finite partitioning of τ , if the probability intensity φ of each epoch class in the partitioning is known, and if the conditional transition probability (given the epoch class) h is also known, the transition intensities u_{ij} are readily determined. The next step is to relate the element properties rigorously to a partitioning of τ , and known functions φ and h .

6.3.4 The Autogenous Epoch Classes

The "autogenous epoch classes" of the network, first described in section 3, are seen to constitute a partitioning of τ . That is, if each autogenous epoch class for each element is represented by an epoch class $\lambda \in \Lambda$, then Λ is a finite set of disjoint subsets of τ whose union is τ . While this is taken axiomatically, it is reasonable to suggest that because an autogenous epoch class represents occurrences which arise spontaneously, (1) every epoch must somewhere have a "spontaneous source," and (2) no epoch can have more than one "spontaneous source." The consistency between viewing autogenous epoch classes as members of a partitioning Λ , and viewing them as representing times whose identity is spontaneously determined within an element becomes more acceptable when it is observed that each of the autogenous epoch classes described in Table I has a probability intensity of occurrence which is well known a priori. In other words, if λ is an autogenous epoch class of e , $\phi(\lambda, i)$ is determined immediately when the definition of the element is given.

From Eq. (6.17), u_{ij} is determined by

$$u_{ij} = \sum_{\lambda \in \Lambda} \phi(\lambda, i) h(\lambda, i, j) \quad (6.18)$$

where

$$h(\lambda, i, j) = \lim_{\Delta t \rightarrow 0} \text{pr}\{Z_{\Delta t} = j \mid \lambda \text{ occurs in } \Delta t \text{ and } Z_0 = i\}.$$

This function can be broken down further through the use of the exogenous epoch classes (and subclasses), and the epoch-trees, which will be described below.

6.3.5 Exogenous Epoch Classes

Each port defines an epoch class representing, under the conventions of Table II (section 3.2.4), either "the times when tasks are offered by (the associate)" or "the times when tasks are requested by (the associate)." These were called exogenous epoch classes. Let ρ_p denote the exogenous epoch class at port p . (It should be noted that every epoch in an exogenous epoch class must also be in an auto-genous epoch class. That is, a "task" may not be "offered" unless action was initiated elsewhere in the network.)

It has been observed (section 5.3.4) that an exogenous epoch may not always produce the same effect as another exogenous epoch in the same class. In different words, the exogenous epoch class ρ_p can be further partitioned to represent distinctions in the action which results from its occurrence. Thus, a finer epoch class for the random switch is "times when tasks are offered by the associate at the input port of a random switch and the switch is set to its upward position," and "times when ... downward position." If ρ_p is the exogenous epoch class at that (input) port, then these two epoch classes represent partitioning subsets of ρ_p .

In general an exogenous epoch class ρ_p at a port p will be partitioned into a set Σ_p of epoch classes. That is

$$\bigcup_{\sigma \in \Sigma_p} \sigma = \rho_p \quad (6.20)$$

and

$$\sigma_1 \cap \sigma_2 = \phi$$

for all σ_1 and σ_2 in Σ_p .

6.3.6 Endogenous Epoch classes

To see how more detailed epoch classes are defined for a network, let λ be an autogenous epoch class of an element e . This epoch class is known to "generate" endogenous epoch classes at a number of the ports of e , which are designated by the endo-event set Q^λ . If $Q^\lambda = \{p_1, p_2, \dots, p_k\}$, then these endogenous epoch classes are synonymous to exogenous epoch classes at the ports connected to p_1, p_2, \dots, p_k —i.e., at ports $p_a = C(p_1), p_b = C(p_2), \dots, p_z = C(p_k)$. (The significance of this "generating" of exogenous epoch classes $\rho_{p_a}, \rho_{p_b}, \dots, \rho_{p_z}$ is that λ is contained in each of these classes.) These epoch classes are partitioned into the sets of epoch classes $\Sigma_{p_a}, \Sigma_{p_b}, \dots, \Sigma_{p_z}$. Consequently there are a number of new epoch classes generated by

$$\nu(\lambda, \sigma_a, \sigma_b, \dots, \sigma_z) = \lambda \cap \sigma_a \cap \sigma_b \cap \dots \cap \sigma_z \quad (6.21)$$

for each $\sigma_a \in \Sigma_{p_a}$, $\sigma_b \in \Sigma_{p_b}$, ..., $\sigma_z \in \Sigma_{p_z}$. In short, there are as many separately distinguishable epoch classes as there are combinations of the partitioning subsets of the exogenous epoch classes.

Furthermore, each of these epoch classes generate endogenous epoch classes at other ports of the elements of which they are a part. These are the elements of the endogenous epoch sets $Q^{\sigma_a}, Q^{\sigma_b}, \dots, Q^{\sigma_z}$ in their respective exogenous events. Intersections of the partitioning subsets continue until no further endogenous epoch classes are generated.

6.4 Epoch Trees and Tree Structured Networks

This process of tracing out the endogenous epoch classes can be made clearer with the aid of Figure 6.1. What is shown are boxes representing the elements, solid lines representing connections, dots representing epoch classes (located at the ports that give rise to them, in the case of exogenous epoch classes), and dotted arrows showing the ports in the endo-event sets of the epochs at their tails. The epoch classes $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ are members of the partitionings $\Sigma_{p_a}, \Sigma_{p_b}, \Sigma_{p_c}, \Sigma_{p_d}$ respectively. Thus, the diagram represents a "smallest" epoch class of the network. Others would result from other combinations, but since the Q^{σ} sets would be different, the "shape" of the diagram would be different for each. Each such epoch class will be called primitive, and to each there will correspond an epoch graph such as Figure 6.2.

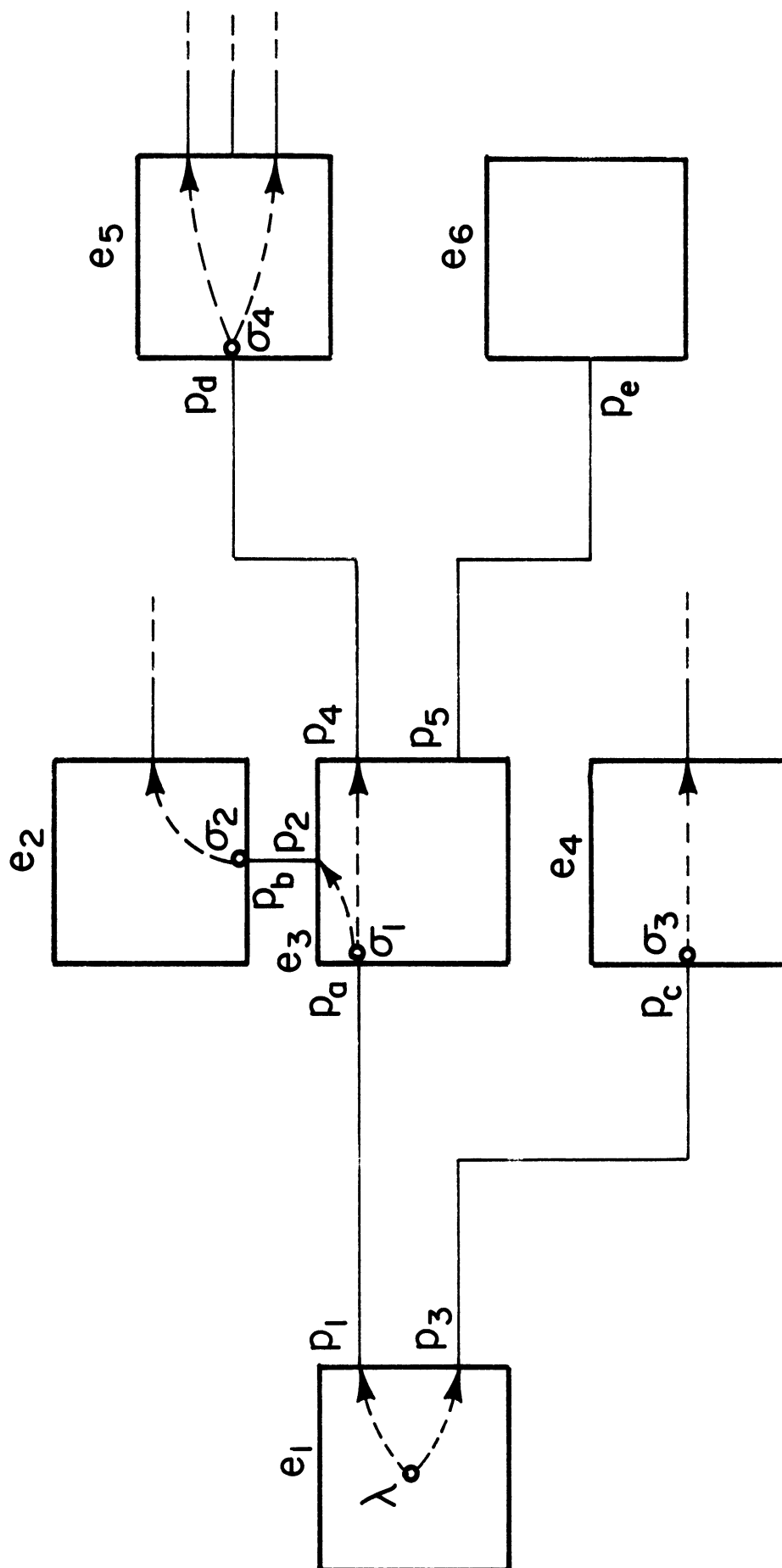


Figure 6.1 Illustrating the "propagation of an epoch."

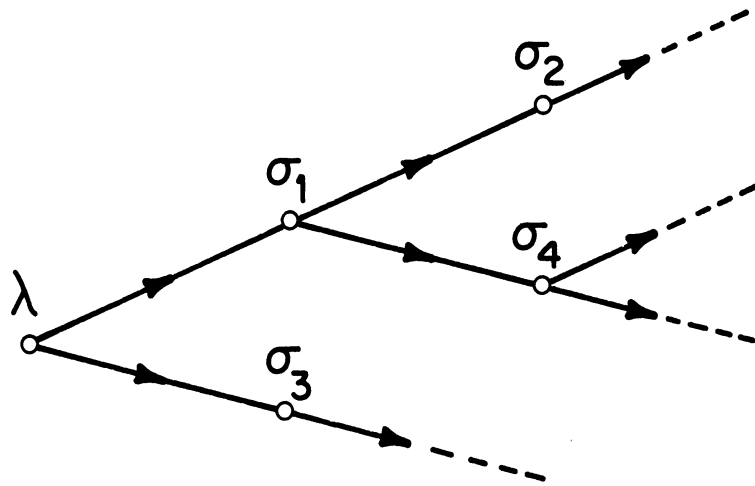


Figure 6.2

An epoch graph fragment corresponding to figure 6.1.

It will be assumed that no graph linking the epoch classes in this manner closes. Hence the epoch graphs are trees, and the networks meeting this restriction will be called tree-structured. This restriction can be relaxed at the expense of further complication of the models. However, such a generalization is beyond our present scope.

A tree-structured network can have a closed diagram. The epoch graph for each primitive epoch class may be open, even though the graph for all of the primitive epoch classes superimposed is closed.

6.6 THE PROBABILITY INTENSITY OF A PRIMITIVE EPOCH CLASS

The set of all primitive epoch classes is itself a partitioning of the epoch set τ . Thus, if we let ν now represent (generically) a primitive epoch class, and let \mathcal{N} now represent the set of all primitive epoch classes, then from Eq. (6.17)

$$u_{ij} = \sum_{\nu \in \mathcal{N}} \phi(\nu, i) h(\nu, i, j), \quad (6.17)$$

where

$$\phi(\nu, i) \triangleq \lim_{\Delta t \rightarrow 0} \frac{\text{pr}\{\nu \text{ occurs in } \Delta t \mid Z_0 = i\}}{\Delta t} \quad (6.12a)$$

represents the (conditional) probability intensity of the occurrence of ν , and where

$$h(\nu, i, j) \triangleq \lim_{\Delta t \rightarrow 0} \text{pr}\{Z_{\Delta t} = j \mid \nu \text{ occurs in } \Delta t \text{ and } Z_0 = i\}. \quad (6.13)$$

If Eq. (6.17) is to be used constructively, $\phi(\nu, i)$ must be determined for all $\nu \in \mathcal{N}$ and all $i \in \mathcal{L}$, and it must specifically be determined

in terms of probabilities and properties of individual elements in the network. The event " ν occurs in Δt " is equivalent to an event reading " $\lambda, \sigma_1, \sigma_2, \dots, \sigma_m$ all occur in Δt ," where $\lambda, \sigma_1, \sigma_2, \dots, \sigma_m$ are the epochs of the tree describing ν . Consequently, φ can be treated like a joint probability, and broken up by use of a conditional chain of the type

$$\begin{aligned} \text{pr}\{\nu \text{ occurs in } \Delta t | Z_0 = i\} &= \text{pr}\{\lambda \text{ occurs in } \Delta t | Z_0 = i\} \text{pr}\{\sigma_1 \text{ occurs in } \Delta t | \lambda \\ &\quad \text{occurs in } \Delta t, Z_0 = i\} \dots \dots \dots \\ &\quad \dots \text{pr}\{\sigma_m \text{ occurs in } \Delta t | \sigma_{m-1}, \dots, \sigma_1, \lambda \\ &\quad \text{all occur in } \Delta t; Z_0 = i\} \end{aligned} \quad (6.22)$$

Let the ordering $\sigma_1, \sigma_2, \dots, \sigma_m$ used in Eq. (6.22) be any ordering such that every epoch class $(\lambda, \sigma_1, \dots, \sigma_m)$ is conditioned only by epoch classes which they do not lead to in the epoch graph. For example, Figure 6.1 has just such an ordering for $\sigma_1, \sigma_2, \sigma_3, \sigma_4, \dots$. (Since the epoch graph is a tree, such an ordering always exists.) Let $(Z_t)_k$ represent the projection of Z_t on the components of Z_t for the element possessing port p_k , and let i_k represent the projection of i upon the corresponding state variables of the element possessing port p_k , $k=1, \dots, m$. Finally, let ρ_k represent the exogenous epoch class at port p_k , and let σ'_k be the immediate predecessor of σ_k on the epoch graph. Then we postulate the following property for the network:

$$\begin{aligned} &\text{pr}\{\sigma_k \text{ occurs in } \Delta t | \sigma_{k-1}, \dots, \sigma_1, \lambda \text{ all occur in } \Delta t; Z_0 = i\} \\ &= \text{pr}\{\sigma_k \text{ occurs in } \Delta t, \sigma'_k \text{ occurs in } \Delta t, Z_0^{(k)} = i_k\} \\ &= \text{pr}\{\sigma_k \text{ occurs in } \Delta t | \rho_k \text{ occurs in } \Delta t, Z_0^{(k)} = i_k\} \end{aligned} \quad (6.23)$$

for $k=2, \dots, m$, and

$$\begin{aligned} & \text{pr}\{\sigma_1 \text{ occurs in } \Delta t | \lambda \text{ occurs in } \Delta t; Z_o = i\} \\ &= \text{pr}\{\sigma_1 \text{ occurs in } \Delta t | \rho_1 \text{ occurs in } \Delta t, Z_o^{(1)} = i_1\} \end{aligned} \quad (6.24)$$

(noting that the predecessor of σ_1 must be λ).

Crudely, what this means is that :

(1) If we know the exogenous epoch class ρ_k , of which σ_k is a part, then a knowledge of the epoch classes which generated the exogenous epoch class ρ_k is not useful in determining the probability of σ_k .

(2) If we know the state of the element of which σ_k is a part, then a knowledge of the state of the rest of the network is similarly useless.

In a similar vein, we also postulate that

$$\text{pr}\{\lambda \text{ occurs in } \Delta t | Z_o = i\} = \text{pr}\{\lambda \text{ occurs in } \Delta t | Z_o^{(0)} = i_0\} \quad (6.25)$$

where $Z_t^{(0)}$ is the projection of Z_t on the process components of Z_t associated with the element defining λ , and i_0 is the projection of i upon the corresponding state variables.

The conditions of these postulates were met by every element proposed in Table I, and are a direct consequence of the so-called "explicitness" of the network. Without this condition, the action of an element would depend upon the character of other elements in the network. It is interesting to note that this property is very similar to the Markovian property, with the (partially ordered) epoch classes of the epoch tree playing the role of the (totally ordered) time axis.

The probabilities of Eq. (6.23) and (6.24) are functions only of σ_k and i_k , since ρ_k can be directly found from σ_k . Thus, let $\psi(\sigma_k, i_k)$ represent this probability for all σ_k in the epoch graph. Also let $\varphi'(\lambda, i_o)$ represent the probability defined by Eq. (6.25). Then Eq. (6.22) becomes

$$\varphi(\nu, i) = \varphi'(\lambda, i_o) \psi(\sigma_1, i_1), \dots, \psi(\sigma_m, i_m), \quad (6.26)$$

and each term is associated with a single element of the network.

6.6 THE CONDITIONAL TRANSITION PROBABILITY

In order to use Eq. (6.17) for the evaluation of the transition intensities u_{ij} , a formulation for determining

$$h(\nu, i, j) \triangleq \lim_{\Delta t \rightarrow 0} \text{pr}\{Z_{\Delta t} = j \mid \nu \text{ occurs in } \Delta t, Z_o = i\} \quad (6.13)$$

for all $i, j \in \mathcal{L}$ and all primitive epoch classes $\nu \in \mathcal{N}$, must be found.

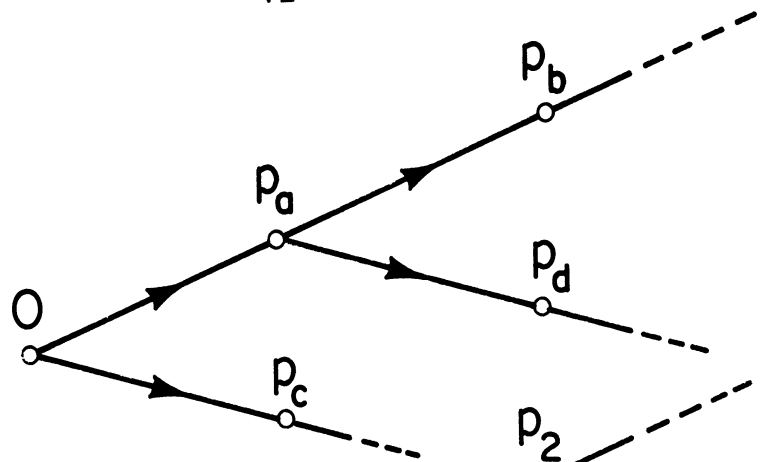
This process is assisted by the creation of a new stochastic process, for which the original Markov chain, $\{Z_t\}$, is a projection. Let $\{\hat{Z}_t, t \geq 0\}$ be this new chain, having components $Z_t^{(1)}, \dots, Z_t^{(n)}, X_t^{(1)}, \dots, X_t^{(\ell)}$, where ℓ is the number of ports in the network. The component $X^{(k)}$ corresponds to the endocondition at the port p_k , $k=1, \dots, \ell$, and is a random variable associated with each port.

There are several additional graphs which it is useful to define,

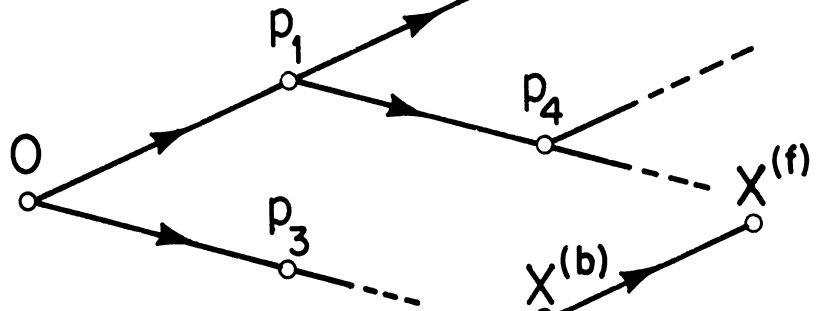
each having the same form as an epoch-graph but with different labeling. If we relabel the nodes of Figure 6.2 with the ports to which the endogenous epochs belong (i. e., the port p such that $\sigma \in \Sigma_p$ for each node σ in the epoch graph) we obtain what will be called the remote-port graph. If we relabel the nodes p of the remote-port graph by the ports to which p is connected, the result is what will be called the near-port graph. (The justification of the terminology should be clear from inspection of Figure 6.1.) Relabeling the nodes with the endocondition at the remote-ports gives an endocondition graph, and the endocontrols at the near-ports gives an endocontrol graph. Finally, labeling the nodes with sets of process components gives a state component graph. The significance attached to each of these graphs, which are illustrated in Figure 6.3 for the example of Figures 6.1 and 6.2, will be explained shortly. The endocontrol w_i at port p_i is a variable which identifies a set of epoch classes $\{\theta_{w_i} : w_i \in \{0, 1, 2, \dots\}\}$ which is another partitioning of the epoch $\rho_{p'_i}$, where p'_i is the (remote) port connected to the (near) port p_i .

There is a special series of properties of networks which were evident in the examples of Section 5. These have to do with the necessity for information about the state, events, or conditions of elements remote from a particular element. It was clear that one could take a fairly myopic view of each element while defining it.

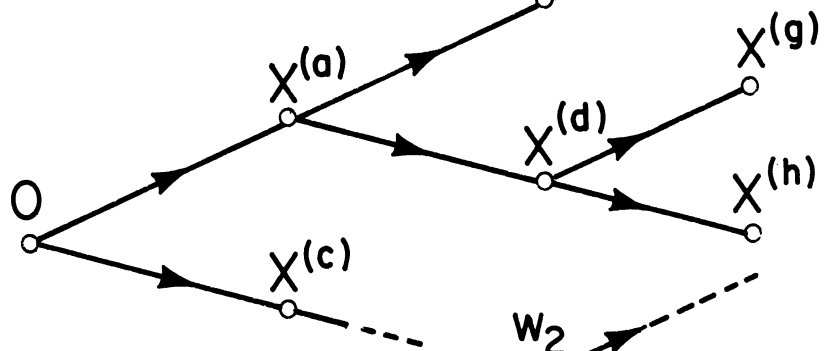
(a) Remote-port Graph



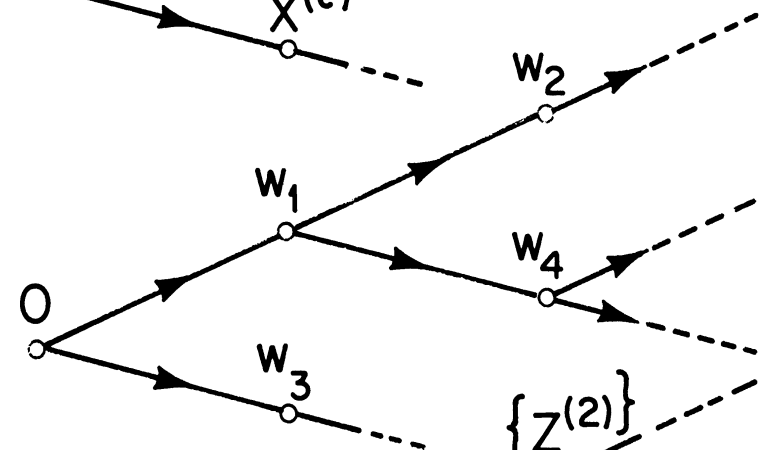
(b) Near-port Graph



(c) Endocondition Graph



(d) Endocontrol Graph



(e) State Component Graph

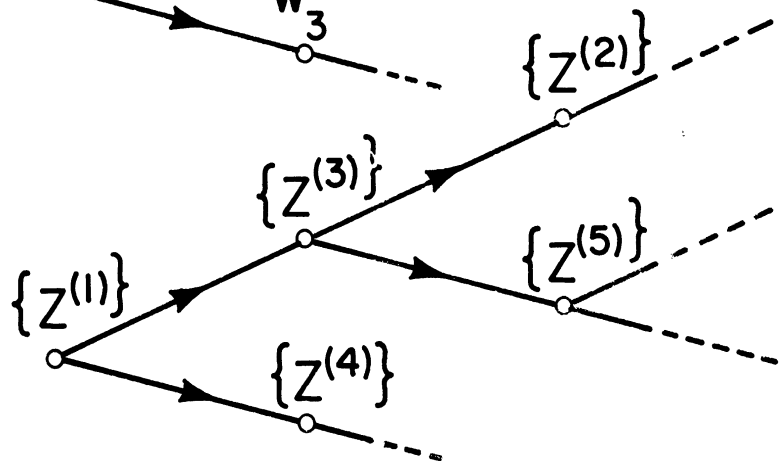


Figure 6.3 Graphs related to the epoch graph.

This shows up as postulated properties for some of the conditional probabilities analogous to those found for epoch classes. The postulated properties are itemized below for a fixed $\nu \in \mathcal{N}$ and a fixed current state Z_0 :

Postulate 1 The network is tree-structured.

Postulate 2 The endoconditions not in the endocondition graph are independent of all other components of \hat{Z}_t .

Postulate 3 The dependence of each endocondition in the endocondition graph upon endoconditions which are not its predecessors in the graph is totally determined by its immediate successors and its state components. Thus, from Figure 6.3

$$\begin{aligned} \text{pr}\{X^{(a)} = x_a \mid X^{(b)} = x_b, X^{(d)} = x_d, X^{(f)} = x_f, X^{(g)} = x_g, \\ X^{(h)} = x_h, Z_0 = z, \nu \text{ occurs in } \Delta t\} \\ = \text{pr}\{X^{(a)} = x_a \mid X^{(b)} = x_b, X^{(d)} = x_d, Z_0^{(3)} = z_3, \\ \nu \text{ occurs in } \Delta t\}. \end{aligned} \quad (6.27)$$

This property is similar to the Markovian property with a partially-ordered set playing the role of the time-axis.

Postulate 4 The dependence of each endocontrol upon endocontrols which are not its successors in the endocontrol graph is similarly totally determined by its immediate predecessor, the state components,

and the corresponding endoconditions. I.e.,
for Figure 6.3

$$\begin{aligned}
 & \text{pr}\{\theta_{w_2} \text{ occurs in } \Delta t \mid \theta_{w_1}, \theta_{w_4}, \theta_{w_3}, \text{ occur in } \Delta t, \\
 & \quad \hat{Z}_0 = z, \nu \text{ occurs in } \Delta t\} \\
 &= \text{pr}\{\theta_{w_2} \text{ occurs in } \Delta t \mid \theta_{w_1} \text{ occurs in } \Delta t, \\
 & \quad X^{(f)} = x_f, Z_0^{(2)} = z_2, \nu \text{ occurs in } \Delta t\} \quad (6.28)
 \end{aligned}$$

Postulate 5 Each state component appears at most once
in the state component graph.

Postulate 6 The target state component $Z_{\Delta t}^{(i)}$, $i=1, 2, \dots, n$,
is determined by the corresponding initial state
components, endocondition, and endocontrols:

$$\begin{aligned}
 & \text{pr}\{Z_{\Delta t}^{(3)} = z'_3 \mid \theta_{w_1}, \theta_{w_2}, \dots \text{ all occur in } \Delta t, \\
 & \quad \hat{Z}_0 = z, Z_{\Delta t}^{(i)} = z'_i \text{ all } i \neq 3\} \\
 &= \text{pr}\{Z_{\Delta t}^{(3)} = z'_3 \mid \theta_{w_1} \text{ occurs in } \Delta t, \\
 & \quad Z_0^{(3)} = z_3, X^{(b)} = x_b, X^{(d)} = x_d\} \quad (6.29)
 \end{aligned}$$

By appropriately chaining all these conditional probabilities,
joint probabilities on $\hat{Z}_{\Delta t}$ and (w_1, \dots, w_ℓ) (conditional on $Z_0 = z_0$)
can be found. An appropriate marginal distribution provides h from
Eq. (6.13), as required. The conditional transition probability func-
tion h is thus one long product of many, many conditional probabilities,
each of which, by postulate, was found to depend upon properties ob-
servable at a single element or its ports. A collection of all those

for the element possessing the autogenous epoch class represents as a product, the true value of the intensity function μ_s^λ used in section 5. The examples were all simple enough so that this complexity did not show itself. (Most of the conditional probabilities were unity.) Some inconsistencies would surely show up if many of the stranger elements had been described.

This is best illustrated by a simple example. Consider the epoch graph of Figure 6.4. We must represent the probability that $(\lambda, \sigma_2, \sigma_4, \sigma_6)$ all occur in Δt , that $X^{(6)} = \alpha_6(s_6)$, that $X^{(2)} = \alpha_2(s_2, \alpha_6(s_6))$, that $X^{(4)} = \alpha_4(s_4)$, that $(\theta_{w_1}^{(1)}, \theta_{w_3}^{(1)}, \theta_{w_5}^{(5)})$ all occur in Δt , that $Z_{\Delta t}^{(1)} = g_1^\lambda(s_1, \alpha_2(s_2, \alpha_6(s_6)), \alpha_4(s_4))$, that $Z_{\Delta t}^{(2)} = g_2^{\sigma_2}(s_2, w_1, \alpha_6(s_6))$, that $Z_{\Delta t}^{(3)} = g_3^{\sigma_4}(s_3, w_3)$, and that $Z_{\Delta t}^{(4)} = g_4^{\sigma_6}(s_4, w_4)$; given $Z_o^{(1)} = s_1$, $Z_o^{(2)} = s_2$, $Z_o^{(4)} = s_3$, and $Z_o^{(4)} = s_4$; where $w_1 = \beta_1^\lambda(s_1, \alpha_2(s_2, \alpha_6(s_6)), \alpha_4(s_4))$, $w_3 = \beta_3^\lambda(s_1, \alpha_2(s_2, \alpha_6(s_6)), \alpha_4(s_4))$, and $w_5 = \beta_5^{\sigma_2}(s_2, w_1, \alpha_6(s_6))$. This probability is the product of the following conditional probabilities, because of the postulates:

- [e₁] (1) $\text{pr}\{\lambda \text{ occurs in } \Delta t | Z_o^{(1)} = s_1\}$
- [e₂] (2) $\text{pr}\{\sigma_2 \text{ occurs in } \Delta t | \rho_2 \text{ occurs in } \Delta t, Z_o^{(2)} = s_2\}$
- [e₃] (3) $\text{pr}\{\sigma_4 \text{ occurs in } \Delta t | \rho_4 \text{ occurs in } \Delta t, Z_o^{(3)} = s_3\}$
- [e₄] (4) $\text{pr}\{\sigma_6 \text{ occurs in } \Delta t | \rho_6 \text{ occurs in } \Delta t, Z_o^{(4)} = s_4\}$
- [e₄] (5) $\text{pr}\{X^{(6)} = \alpha_6(s_6) | \sigma_6 \text{ occurs in } \Delta t, Z_o^{(4)} = s_4\}$
- [e₂] (6) $\text{pr}\{X^{(2)} = \alpha_2(s_2, \alpha_6(s_6)) | \sigma_2 \text{ occurs in } \Delta t, X^{(6)} = \alpha_6(s_6), Z_o^{(2)} = s_2\}$

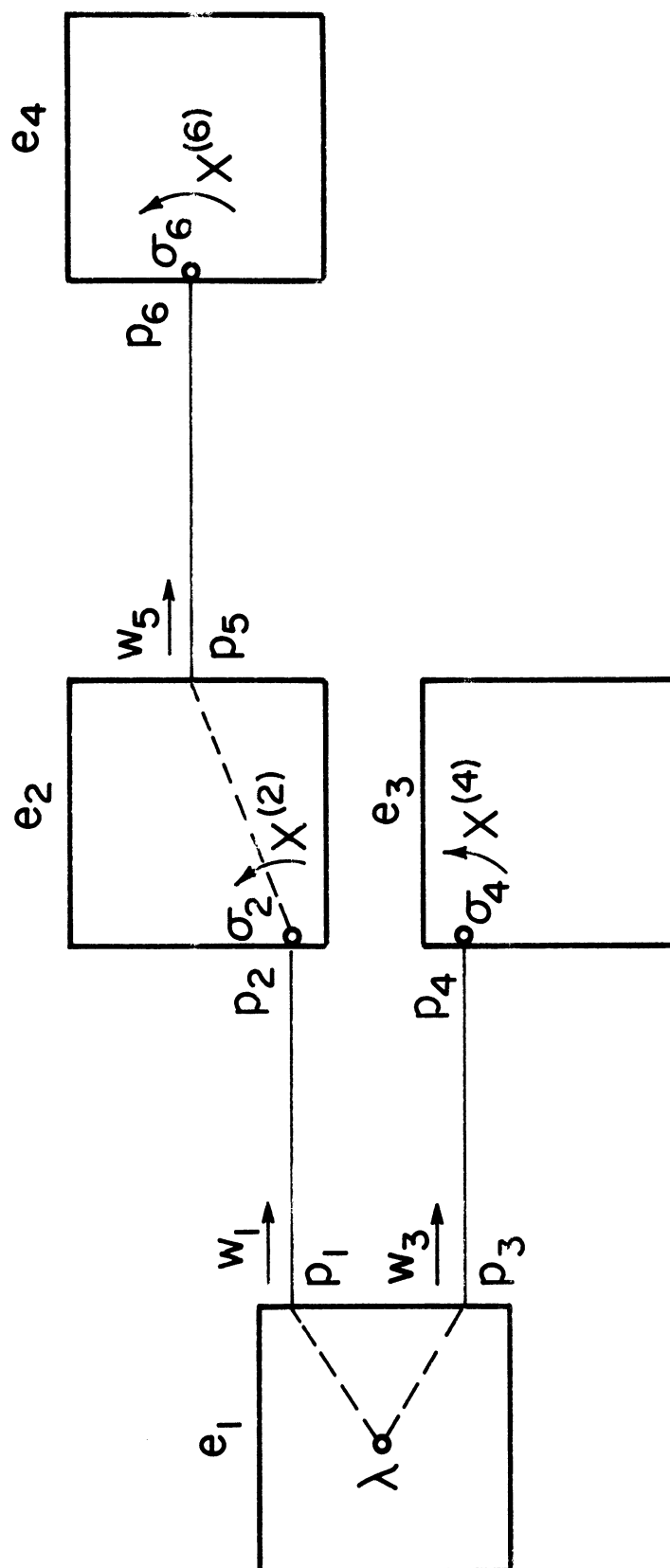


Figure 6.4 An epoch graph.

$$[e_3] \quad (7) \quad \text{pr}\{X^{(4)} = \alpha_4(s_4) | \sigma_4 \text{ occurs in } \Delta t, Z_o^{(3)} = s_3\}$$

$$[e_1] \quad (8) \quad \text{pr}\{\theta_{w_1}^{(1)} \text{ and } \theta_{w_3}^{(3)} \text{ both occur in } \Delta t, Z_{\Delta t}^{(1)} = g_1^\lambda(\dots)$$

$$| \lambda \text{ occurs in } \Delta t, X^{(2)} = \alpha_2(\dots),$$

$$X^{(4)} = \alpha_4(s_4), Z_o^{(1)} = s_1\}$$

$$[e_2] \quad (9) \quad \text{pr}\{\theta_{w_5}^{(5)} \text{ occurs in } \Delta t, Z_{\Delta t}^{(2)} = g_2^\lambda(\dots)$$

$$| \sigma_2 \text{ and } \theta_{w_1}^{(1)} \text{ both occur in } \Delta t,$$

$$X^{(6)} = \alpha_6(s_6), Z_o^{(2)} = s_2\}$$

$$[e_3] \quad (10) \quad \text{pr}\{Z_{\Delta t}^{(3)} = g_3^{\sigma_4}(s_3, w_3) | \sigma_4 \text{ occurs in } \Delta t, Z_o^{(3)} = s_3\}$$

$$[e_4] \quad (11) \quad \text{pr}\{Z_{\Delta t}^{(4)} = g_4^{\sigma_6}(s_4, w_4) | \sigma_6 \text{ occurs in } \Delta t, Z_o^{(4)} = s_4\}$$

The left-most margin indicates which element totally determines the given probability.

From this we see that the intensity function μ_s^λ used in section 5 represents all those terms associated with the element having the autogenous event. Thus, reverting to the previous, more general, notation

$$\begin{aligned} \mu_s^\lambda &= \lim_{\Delta t \rightarrow 0} \text{pr}\{\lambda \text{ occurs in } \Delta t | Z_o^{(0)} = s_o\} \\ &\quad \times \text{pr}\{\theta_{w_Q} \text{ all occur in } \Delta t, Z_{\Delta t}^{(0)} = g_s^\lambda(x_Q) \\ &\quad | \lambda \text{ occurs in } \Delta t, X^Q = x_Q\} / \Delta t \end{aligned} \quad (6.30)$$

where θ_{w_Q} is a shorthand for $\theta_{w_1}, \theta_{w_2}, \dots$, where p_1, p_2, \dots are in Q ; and where x_Q is shorthand for x_1, x_2, \dots the exoconditions at the ports in Q .

We also see that the probability function $\pi_s^{\sigma_k}$ represents all those terms associated with the element having σ_k for an exogenous epoch at port p_k . Thus

$$\begin{aligned} \pi_s^{\sigma_k} &= \lim_{\Delta t \rightarrow 0} \text{pr}\{\sigma_k \text{ occurs in } \Delta t \mid \rho_p \text{ occurs in } \Delta t, Z_o^{(k)} = s_k\} \\ &\cdot \text{pr}\{X^{(k)} = \alpha_{s_k}^{\sigma_k}(x_Q) \mid \sigma_k \text{ occurs in } \Delta t, X^{(Q)} = x_Q, Z_o^{(k)} = s_k\} \\ &\cdot \text{pr}\{\theta_{w_Q} \text{ all occur in } \Delta t, (Z_{\Delta t})_k = g_{s_k}^{\sigma_k}(x_Q, y_{p_k}) \\ &\mid \sigma_k \text{ and } \theta_{w_{p_k}} \text{ both occur in } \Delta t, X^Q = x_Q, Z_o^{(k)} = s_k\} \quad (6.31) \end{aligned}$$

It should be carefully noted that neither μ_s^λ nor π_s^σ represent meaningful conditional probability intensities or probabilities, because the conditions are not "chained." However, they do represent a product of legitimate conditional probabilities, and the product of all such probabilities for the primitive epoch class ν (as shown in an epoch graph) is very meaningful. It is a large joint probability whose sums are over appropriate variables results in the determination of $\varphi(\nu, i)h(\nu, i, j)$.

7. THE TRANSLATION OF NETWORKS TO EQUILIBRIUM EQUATIONS

The objective of this report has been to provide the capabilities needed to specify a Markovian system in pictorial language form, and then to systematically translate the specification (we called it a diagram) into a straightforward description of the equilibrium equations for the Markov chain which models it. It was agreed that the matrix of transition intensities (u_{ij}) would suffice, provided its index set was suitably identified with the states of the system.

The problem has, up to now, been almost entirely representational. However, with the background provided, we can finally take an operational view, and develop the procedure. An operation called consolidation will be central to the procedure for translation, and the major part of this section will be devoted to its description. Finally, its use for the task of translation will be presented.

7.1 CONSOLIDATION OF NETWORKS

The operation of network consolidation consists of forming a new network which is equivalent^{*} to the old, but for which one connection (and its meaning) has been absorbed into the elements. There are two cases, illustrated in Figure 7.1. Either the connection to be

* We will say the two networks, N and \bar{N} , are equivalent if they describe the same Markov chain (in the same state space).

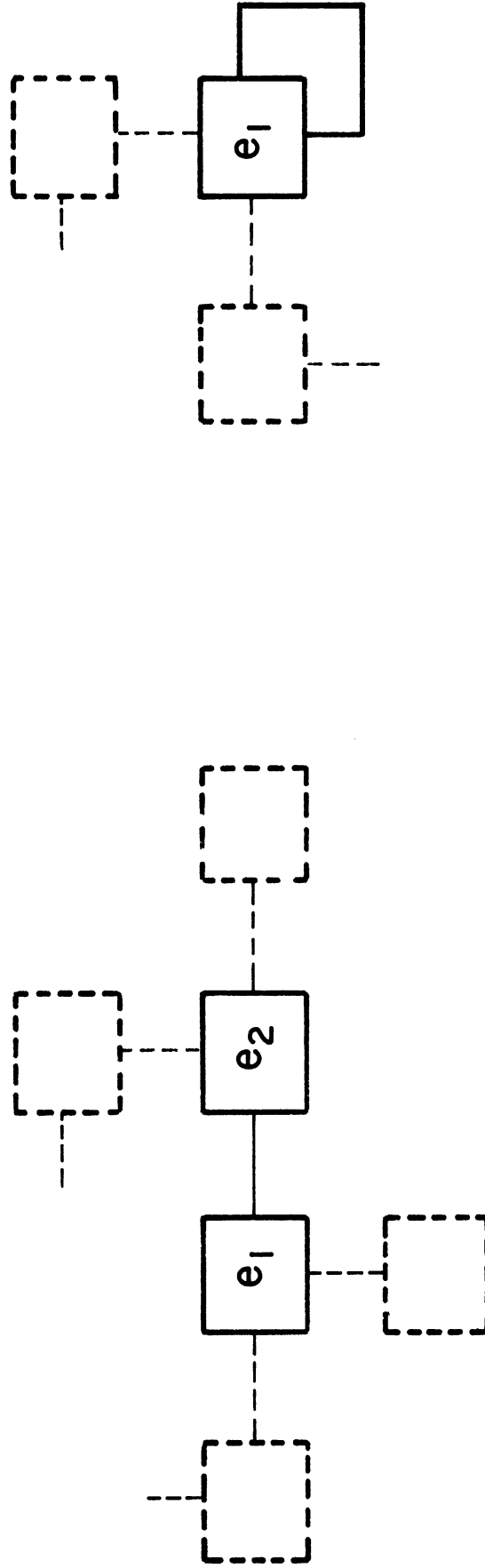


Figure 7.1

Illustrating two cases for consolidation.

absorbed joins ports of two different elements, or it joins two ports of the same element. Symbolically, if the new network is \bar{N} and the old is N , and if one of the ports involved is p_1 , the consolidation operation \mathcal{C} can be written

$$\bar{N} = \mathcal{C}(N, p_1). \quad (7.1)$$

The other port (p_2) involved is obviously found from the connection function C :

$$p_2 = C(p_1). \quad (7.2)$$

As a result of the absorption of the connection, new elements must be formed. In the first case, the elements e_1 and e_2 will be "consolidated" into a single element \bar{e} having two less ports than the two elements had together. In the second case e_1 will be replaced by a new element \bar{e} which has two less ports than e_1 . All elements except these remain unchanged, as well as all connections other than the one between p_1 and p_2 .

Figure 7.2 illustrates the consolidated networks. Symbolically, if we let $\bar{N} = (\bar{\mathcal{E}}, \bar{C})$ then

$$\bar{\mathcal{E}} = (\mathcal{E} - \{e_1, e_2\}) \cup \bar{e}. \quad (7.3)$$

(We adopt the convention of defining $e_2 \triangleq e_1$ for case 2.) Every connection other than the one being absorbed remains the same, so that the new connection function is simply a restriction of C to the set

$$\bar{\mathcal{P}} = \mathcal{P} - \{p_1, p_2\}. \quad (7.4)$$

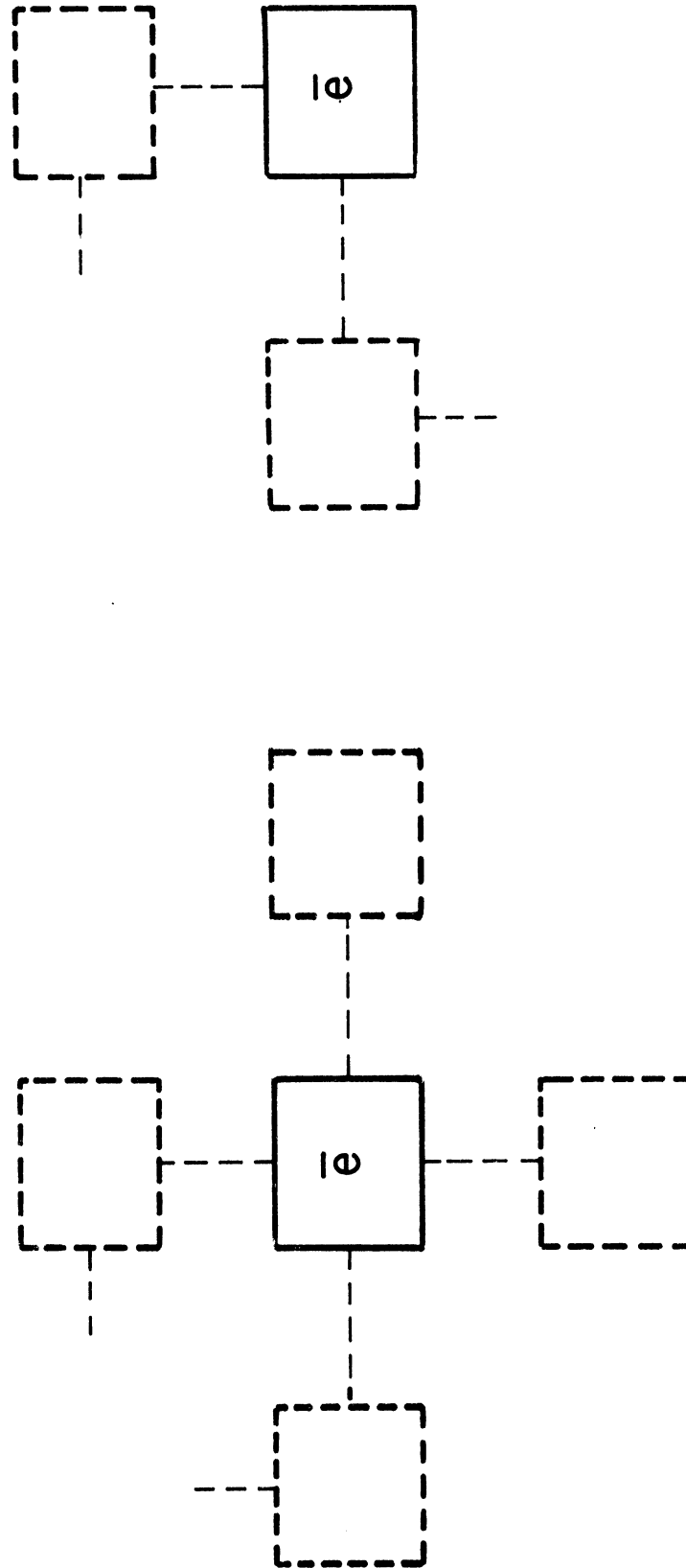


Figure 7.2

Illustrating the "consolidated" networks for figure 7.1

I. e. ,

$$\bar{C}(p) = C(p) \text{ for all } p \in \mathcal{P} - \{p_1, p_2\}. \quad (7.5)$$

Thus, the principal operation in network consolidation is the operation which determines the element \bar{e} . This operation is called element consolidation. In case 1 we write that $\bar{e} = \mathcal{C}'(e_1, e_2, p_1, p_2)$, and in case 2 we write that $\bar{e} = \mathcal{C}''(e_1, p_1, p_2)$, where the operations \mathcal{C}' and \mathcal{C}'' represent the element consolidation. To define these operations it is clearly necessary, and sufficient, to define a port set \bar{P} , a state set \bar{S} , an auto-event set $\bar{\Xi}$, and an exo-event set \bar{Z} such that $\bar{e} \triangleq (\bar{P}, \bar{S}, \bar{\Xi}, \bar{Z})$ and the network \bar{N} defines the same Markov chain as N . Only case 1 will be considered here. Case 2 follows trivially from this procedure for tree-structured networks.

The port set \bar{P} is the easiest to derive. Since every port which was a port of either e_1 or e_2 must be a port of \bar{e} , with the exception of p_1 and p_2 , \bar{P} is obviously

$$\bar{P} = (P_1 \cup P_2) - \{p_1, p_2\}, \quad (7.6)$$

where P_1 and P_2 are the port sets of e_1 and e_2 respectively. This permits every connection other than the one being absorbed to remain unchanged.

More will be said about the state set \bar{S} later. However, it is clear that if the "condition" of the new element is to be described, the values of the state variables of both elements must be known. Therefore

S is at most the set $S_1 \times S_2$ of pairs^{*}, or

$$\bar{S} \subseteq S_1 \times S_2. \quad (7.7)$$

This is sufficient knowledge of \bar{S} to proceed with the descriptions of \bar{E} and \bar{Z} .

7.2 CONSOLIDATED AUTOGENOUS EVENTS

While the consolidation of the auto-events is a highly detailed operation, in principle it is quite straightforward. Each port of an element identifies, under appropriate circumstances, an endocondition function and an endocontrol function. In Section 3 we had used the schematic reproduced in Figure 7.3 to illustrate this state of affairs, where v represents an endocondition, and w represents an endocontrol. Under other circumstances, we have said that the behavior of the element is frequently dependent upon exoconditions and exocontrols, which are schematically shown as x and y in Figure 7.3. The endocontrols and the endoconditions were determined, as required, by the endocontrol and endocondition functions which defined the element. The exocontrols and exoconditions were treated as variables

* Unless otherwise stated, the Cartesian product operator will be taken in an associative form so that $(A \times B) \times C = A \times (B \times C)$. Thus, if $S_1 = S_a \times S_b$ and $S_2 = S_c \times S_d$, then $(a, b, c, d) \in S_1 \times S_2$ if $(a, b) \in S_1$ and $(c, d) \in S_2$. Strictly, one otherwise means $((a, b), (c, d)) \in S_1 \times S_2$ under these circumstances.

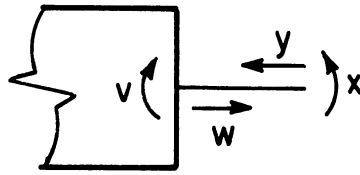


Figure 7.3 Schematic of port variables.

upon which these functions depended, and were to be determined by the network external to the element—in particular, by the endocontrol and endocondition of the associate at the port. This situation is indicated schematically in Figure 7.4. Evidently a simple substitution, representing x_2 instead by the function defining v_1 , y_2 by the function defining w_1 , x_1 by the function defining v_2 , and y_1 by the function defining w_2 . The variables x_1, x_2, y_1, y_2 would then cease to exist, and everything which depended upon them would depend instead upon the variables upon which v_1, v_2, w_1, w_2 depended. Furthermore, the functions defining v_1, v_2, w_1 and w_2 could now be absorbed by defining new functions which were compositions of the

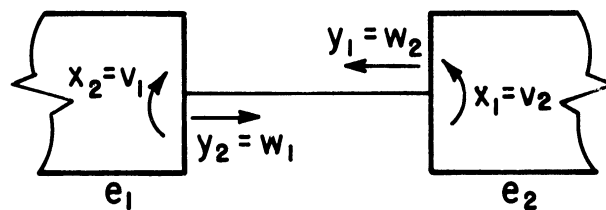


Figure 7.4 Relationships of port variables in connected elements.

functions defining v_1, v_2, w_1 and w_2 with the functions which originally depended upon x_1, x_2, y_1 and y_2 . Thus all variables or functions identified with the two ports would disappear from view.

The application of this simple principle is practically all that is needed to consolidate the elements, and, in particular, the auto-events. However, considerable detail ensues from the detailed structure of the element and the auto-event definitions, and the ways in which the endocondition and endocontrol functions are buried in this structure.

The auto-event sets of elements e_1 and e_2 are the sets

$$\begin{aligned}\Xi_1 &= \{\xi^\lambda : \lambda \in \Lambda_1\} \\ \Xi_2 &= \{\xi : \lambda \in \Lambda_2\}\end{aligned}\tag{7.8}$$

respectively, while that of \bar{e} is

$$\bar{\Xi} = \{\bar{\xi}^\lambda : \lambda \in \bar{\Lambda}\}\tag{7.9}$$

where Λ_1 and Λ_2 are disjoint, and $\bar{\Lambda}$ is a new set of epoch classes associated with the consolidated element. Clearly, since the underlying Markov chain for the network must not change under this operation, each epoch in Λ_1 and Λ_2 must have a counterpart in the epoch set $\bar{\Lambda}$. Put in other words, each auto-event of the elements e_1 and e_2 represent "actions" of the consolidated element. At first blush, one might expect that the epochs in $\bar{\Lambda}$ would correspond one-for-one to the epochs in the union of Λ_1 and Λ_2 . This is, however, not the case.

As it happens some of these epochs may split into several new ones, so that $\bar{\Lambda}$ is not necessarily the union of Λ_1 and Λ_2 . To see why this is so, consider an epoch class $\lambda_1 \in \Lambda_1$ which produces an endogenous epoch class of e_1 at port p_1 . Since this is an exogenous epoch class at p_2 (because of the connection between p_1 and p_2), we must look to the exogenous events of e_2 to fully explore what happens. Here

$$\mathbb{Z}_2(p) = \{\xi : \sigma \in \Sigma_p\}, \quad p \in P_2, \quad (7.10)$$

and if Σ_{p_2} contains more than one member, there will be several consequences within e_2 resulting from the same auto-epoch in e_1 .

To be more concrete, suppose e_1 is a server and e_2 is a random switch (see Figure 7.5). Then, when a service completion occurs in e_1 ,

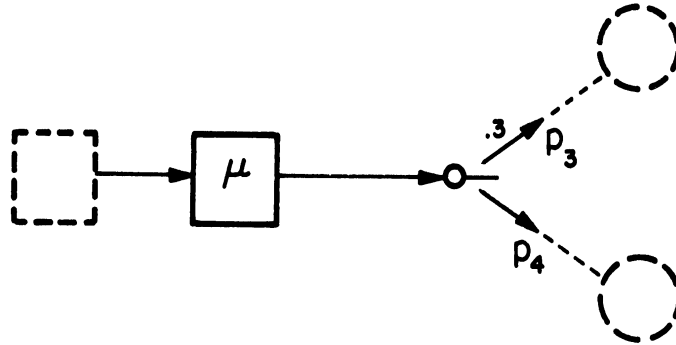


Figure 7.5 Illustrating epoch-splitting.

either an endo-event occurs at port p_3 or it occurs at p_4 . For the consolidated element, there would be two auto-epoch classes: one which offered a task out port p_3 and one which offered a task out port

p_4 . This illustration also shows that, in general, the exo-events at ports p_1 and p_2 will also be involved in the operation of deriving the consolidated auto-events.

Recall that every auto-event ξ^λ of an element e is defined by

$$\xi_s^\lambda = (Q^\lambda, \beta_s^\lambda, g_s^\lambda, \mu_s^\lambda), \lambda \in \Lambda_e, s \in S_e$$

and every exo-event ζ^σ ,

$$\zeta_s^\sigma(p) = (Q^\sigma, \alpha_s^\sigma, \beta_s^\sigma, g_s^\sigma, \pi_s^\sigma), \sigma \in \Sigma_p, p \in P_e, s \in S_e$$

and that Q^λ represents the set of ports having endo-epoch classes generated by the epoch class λ . The role of the endo-epoch sets Q^λ and Q^σ is more clearly seen with the aid of the epoch-graphs of Figure 7.6, which illustrates two possible situations. In case (a) element e_1 possesses an auto-event whose endo-epoch set contains the ports p_1, p_a , and p_b while e_2 possesses an exo-event at port p_2 containing p_c and p_d . In other words $Q^\lambda = \{p_1, p_a, p_b\}$ and $Q^\sigma = \{p_c, p_d\}$. Clearly, the consolidated auto-event corresponding to this is one which has an endo-event set consisting of p_a, p_b, p_c , and p_d . On the other hand, in case (b) the endo-epoch set for λ in e_1 does not contain p_1 , so that the exo-event at port p_2 will not be excited. The consolidated auto-event must have an endo-epoch set consisting of only p_a and p_b . In this latter case, the consolidated auto-event must, in some sense, be simply an extension of the auto-event of e_1 , since no exo-events at p_2 are involved.

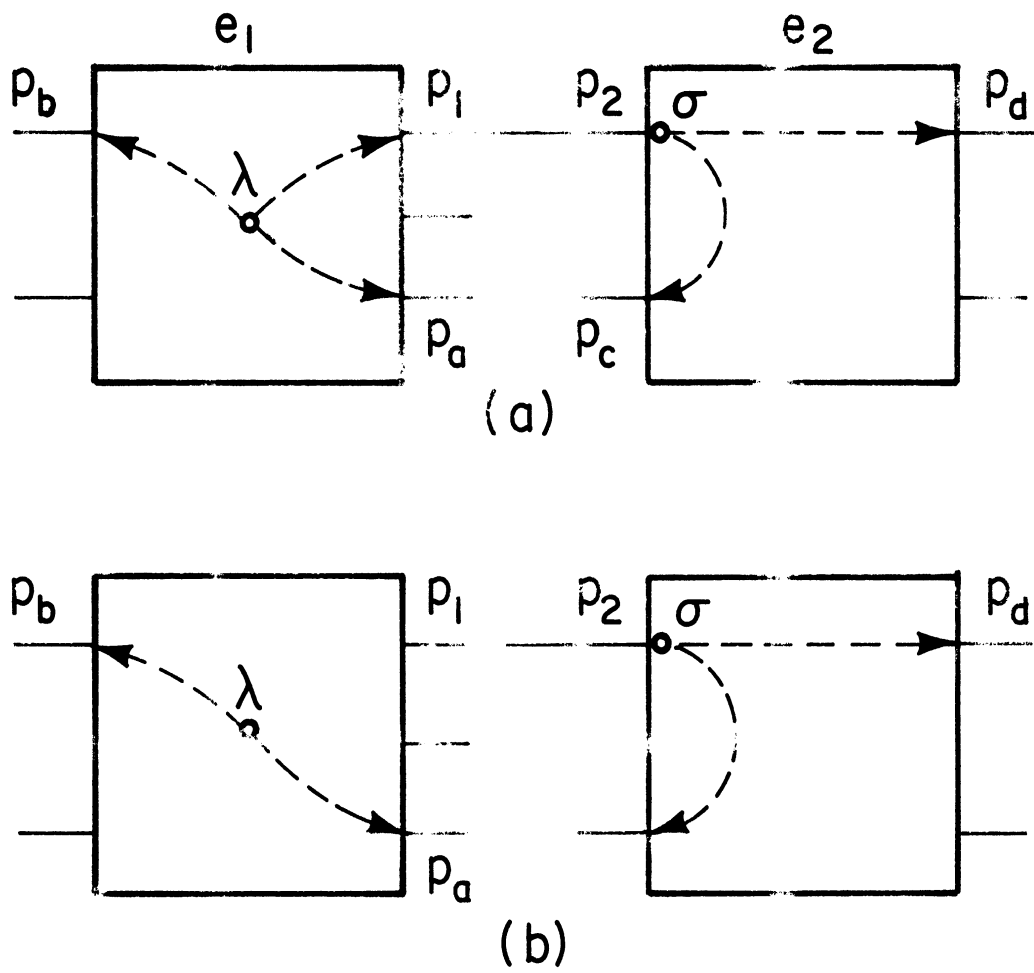


Figure 7. 6

Illustrating epochs and their influence.

From these examples, we can distinguish the following two useful subsets of the epoch set Λ_1 :

- (1) Those epochs λ for which $p_1 \in Q^\lambda$. Call the set of these Λ'_1 :

$$\Lambda'_1 = \{\lambda: \lambda \in \Lambda_1, p_1 \notin Q^\lambda\} \quad (7.11)$$

- (2) Those epochs λ for which $p_1 \in Q^\lambda$. Call the set of these Λ''_1 :

$$\Lambda''_1 = \{\lambda: \lambda \in \Lambda_1, p_1 \in Q^\lambda\}. \quad (7.12)$$

Notice that Λ'_1 and Λ''_1 together form a partition of Λ_1 . A similar partition $\{\Lambda'_2, \Lambda''_2\}$ can be found for Λ_2 .

Case (1)

Let

$$\bar{\xi}_s^\lambda = (\bar{Q}^\lambda, \bar{\beta}_s^\lambda, \bar{g}_s^\lambda, \bar{\mu}_s^\lambda) \quad (7.13)$$

for all $s \in \bar{S}$ and $\lambda \in \bar{\Lambda}$. For case (1), above, each of the auto-events of e_1 in Λ'_1 does not involve any of the exo-events of e_2 . Consequently, the consolidated element \bar{e} will have one auto-event for each $\lambda \in \Lambda'_1$, and these events will have precisely the same influence on the ports of \bar{e} as did e_1 . Therefore, let Λ'_1 be contained in $\bar{\Lambda}$, and write

$$\bar{Q}^\lambda = Q^\lambda, \quad (7.14)$$

for all $\lambda \in \Lambda'_1 \subseteq \bar{\Lambda}$. Further, recall that if $Q^\lambda = \{p_a, p_b, \dots, p_z\}$, then $\beta_{S_1}^\lambda$ is a function on $I_{p_a} \times I_{p_b} \times \dots \times I_{p_z}$ to $I_{p_a} \times I_{p_b} \times \dots \times I_{p_z}$, representing the endo-controls at the ports of Q^λ , as functions of the exoconditions at the same ports. These are all unchanged by the consolidation so that

we write

$$\bar{\beta}^{\lambda}(s_1, s_2) = \beta^{\lambda} s_1 \quad (7.15)$$

for all $s_1 \in S_1$, $(s_1, s_2) \in \bar{S} \subseteq S_1 \times S_2$, and $\lambda \in \Lambda'_1 \subseteq \bar{\Lambda}$. Still further, \bar{g}_s^{λ} is the target state function, which is a function of the exoconditions at the ports of Q^{λ} , to the set $\bar{S} \subseteq S_1 \times S_2$. The second state variable will be uninfluenced by the event so that

$$\bar{g}^{\lambda}(s_1, s_2) = (g^{\lambda}_{s_1, s_2}) \quad (7.16)$$

for all $s_1 \in S_1, \dots$, etc. Finally, the probability intensity of the epoch is also unchanged, so we write

$$\bar{\mu}^{\lambda}(s_1, s_2) = \mu^{\lambda} s_1 \quad (7.17)$$

for all $s_1 \in S_1, \dots$, etc. This defines the auto-events for all $\lambda \in \Lambda'_1$. An interchange of the roles of e_1 and e_2 will define the auto-events $\bar{\xi}^{\lambda}$ for all $\lambda \in \Lambda'_1$.

Case (2)

For case (2), things are a little more complicated. Besides the situation illustrated in Figure 7.6, one can encounter the situation (the "random exo-event") illustrated schematically in Figure 7.7. In this case Σ_{p_2} (see Eq. (7.10)) consists of two elements σ_a and σ_b , and as a result the epoch λ is split into two epochs in the consolidated element (with a probability intensity which is that of λ split between the two new

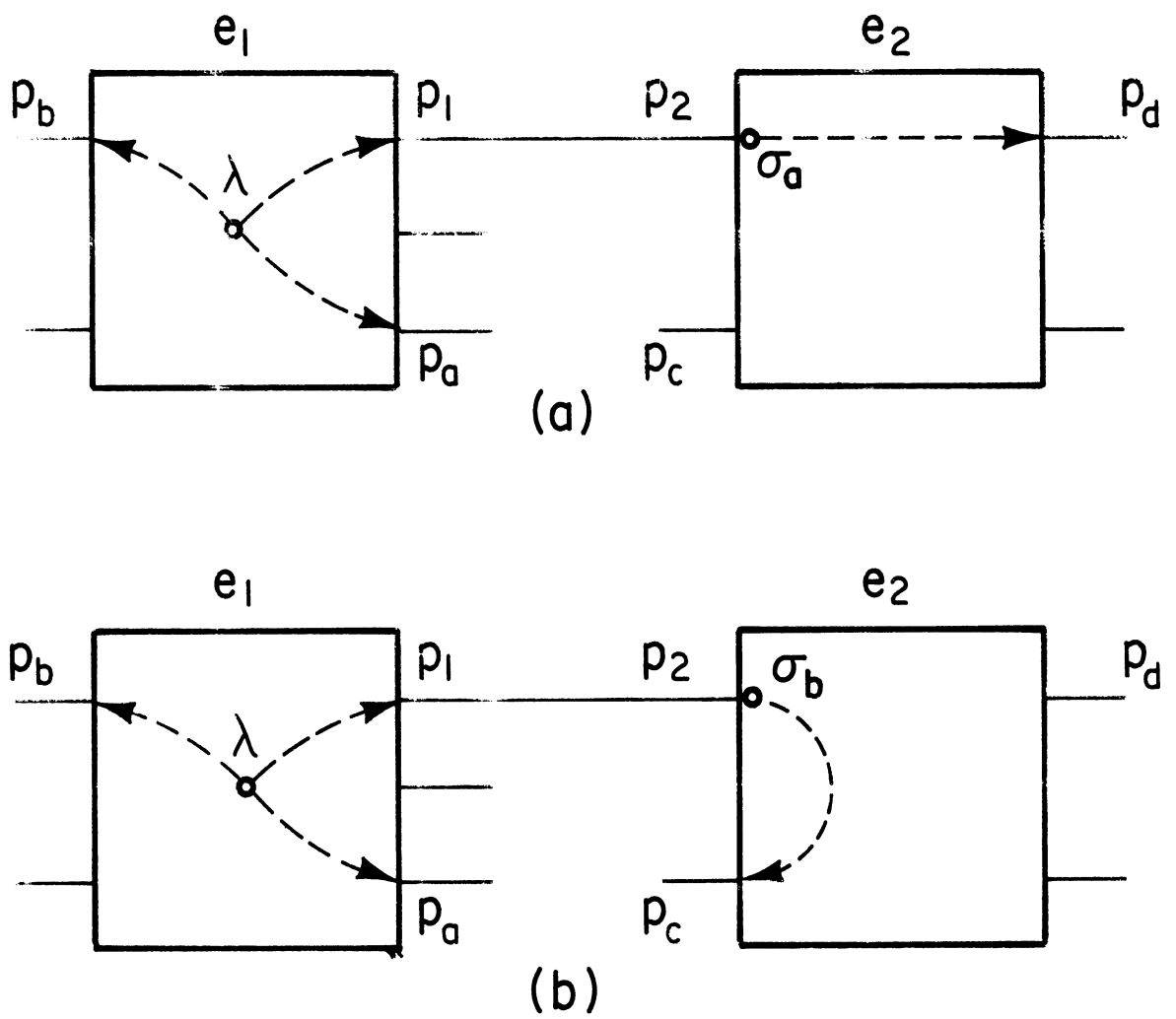


Figure 7. 7

Illustrating epochs and their influence.

epochs). One of these has endo-event set $\{p_a, p_b, p_d\}$ and the other has endo-event set $\{p_a, p_b, p_c\}$. In general each epoch class in Λ''_1 will result in as many epoch classes in $\bar{\Lambda}$ as there are elements in Σ_{p_2} . We can thus name these new epochs by the double $(\lambda, \sigma) \in \Lambda''_1 \times \Sigma_{p_2}$ and let $\bar{\Lambda} \supseteq \Lambda''_1 \times \Sigma_{p_2}$. Similarly, each epoch in Λ''_2 will result in epochs corresponding to the elements of $\Lambda''_2 \times \Sigma_{p_1}$. Furthermore, there can be no epochs of \bar{e} other than those mentioned so that we say

$$\bar{\Lambda} = \Lambda''_1 \cup \Lambda''_2 \cup (\Lambda''_1 \times \Sigma_{p_2}) \cup (\Lambda''_2 \times \Sigma_{p_1}). \quad (7.18)$$

We now define the $\bar{\xi}^\lambda$ for $\lambda \in (\Lambda''_1 \times \Sigma_{p_2})$. Observing Figure 7.7, the endo-epoch set is easily defined as

$$\bar{Q}^{(\lambda, \sigma)} = Q^\lambda \cup Q^\sigma - \{p_1\} \quad (7.19)$$

for all $\lambda \in \Lambda''_1$, and $\sigma \in \Sigma_{p_2}$. Then, for the example illustrated,

$$\begin{aligned} \bar{Q}^{(\lambda, \sigma_a)} &= \{p_a, p_b, p_d\} \\ \bar{Q}^{(\lambda, \sigma_b)} &= \{p_a, p_b, p_c\}. \end{aligned} \quad (7.20)$$

The consolidated endo-control function $\bar{\beta}_{(s_1, s_2)}^{(\lambda, \sigma)}$ must be a function on $\bar{J}^{(\lambda, \sigma)}$ to $\bar{J}^{(\lambda, \sigma)}$, where $\bar{J}^{(\lambda, \sigma)} = I_{p_a} \times \dots \times I_{p_z} \times I_{p'_a} \times \dots \times I_{p'_z}$, $\{p_a, \dots, p_z\} = Q^\lambda - \{p_1\}$, $\{p'_a, \dots, p'_z\} = Q^\sigma$, and $I_p = \{0, 1, 2, \dots\}$ for all $p \in \bar{P}$. Let us define the components of $\bar{\beta}_{(s_1, s_2)}^{(\lambda, \sigma)}$ by

$$\bar{\beta}_{(s_1, s_2)}^{\lambda, \sigma} = (\bar{\beta}_{p_a}, \dots, \bar{\beta}_{p_z}, \bar{\beta}_{p'_a}, \dots, \bar{\beta}_{p'_z}), \quad (7.21)$$

the components of $\beta^\lambda_{s_1}$ by

$$\beta^\lambda_{s_1} = (\beta_{p_a}, \dots, \beta_{p_z}, \beta_{p_1}), \quad (7.22)$$

the components of $\beta^\lambda_{s_2}$ by

$$\beta^\sigma_{s_2} = (\beta_{p'_a}, \dots, \beta_{p'_z}, \beta_{p_2}), \quad (7.23)$$

and the endocondition function at port p_2 by $\alpha^\sigma_{p_2}$, for all $\lambda \in \Lambda''_1$ and all $\sigma \in \Sigma_{p_2}$. Then the first group of components of $\bar{\beta}^{(\lambda, \sigma)}_{(s_1, s_2)}$ is found from $\beta^\lambda_{s_1}$ by replacing x_{p_1} by $\alpha^\sigma_{p_2}(x_{p'_a}, \dots, x_{p'_z})$ as follows

$$\bar{\beta}_{p_{p_a}, \dots, p_z, p'_a, \dots, p'_z}^{(\lambda, \sigma)} = \beta_{p_{p_a}, \dots, p_z, \alpha^\sigma_{p_2}(x_{p'_a}, \dots, x_{p'_z})}^\lambda \quad (7.24)$$

for all $p \in \{p_a, \dots, p_z\}$ and all $(s_1, s_2) \in \bar{S}$, $\lambda \in \Lambda''_1$, and $\sigma \in \Sigma_{p_2}$.

The second group of components of $\bar{\beta}^{(\lambda, \sigma)}_{(s_1, s_2)}$ is found from $\beta^\sigma_{s_2}$ by replacing the exocontrol y_{p_2} by $\beta_{p_1, p_a, \dots, p_z, \alpha^\sigma_{p_2}(x_{p'_a}, \dots, x_{p'_z})}$, so that

$$\bar{\beta}_{p_{p_a}, \dots, p_z, p'_a, \dots, p'_z}^{(\lambda, \sigma)} = \beta_{p_{p'_a}, \dots, p_{p'_z}, \beta_{p_1, p_a, \dots, p_z, \alpha^\sigma_{p_2}(x_{p'_a}, \dots, x_{p'_z})}}^\sigma \quad (7.25)$$

for all $p \in \{p'_a, \dots, p'_z\}$ and all $(s_1, s_2) \in \bar{S}$, $\lambda \in \Lambda''_1$ and $\sigma \in \Sigma_{p_2}$.

The formation of the target state function $\bar{g}_{(s_1, s_2)}^{(\lambda, \sigma)}$ proceeds similarly. Let the components of $\bar{g}_{(s_1, s_2)}^{(\lambda, \sigma)}$ be defined by

$$\bar{g}_{(s_1, s_2)}^{(\lambda, \sigma)} = (\bar{g}_1, \bar{g}_2) \quad (7.26)$$

for all $\lambda \in \Lambda''_1$, and all $\sigma \in \Sigma_{p_2}$. Then

$$\begin{aligned} \bar{g}_1(x_{p_a}, \dots, x_{p_z}, x_{p'_a}, \dots, x_{p'_b}) &= g_{s_1}^\lambda(x_{p_a}, \dots, x_{p_z}, \alpha_{p_2}^\sigma(x_{p'_a}, \dots, \\ &\quad x_{p'_z})) \end{aligned} \quad (7.27)$$

and

$$\begin{aligned} \bar{g}_2(x_{p_a}, \dots, x_{p_z}, x_{p'_a}, \dots, x_{p'_b}) &= g_{s_2}^\sigma(x_{p'_a}, \dots, x_{p'_z}, \beta_{p_1}(x_{p_a}, \dots, \\ &\quad x_{p_z}, \alpha_{p_2}^\sigma(x_{p'_a}, \dots, x_{p'_z}))) \end{aligned}$$

for all $(s_1, s_2) \in S$, etc. .

The probability intensity of the epoch (λ, σ) is found by multiplying the probability intensity function $\mu_{s_1}^\lambda$ of the autogenous event ξ^λ by the conditional probability function $\pi_{s_2}^\sigma$ of ζ^σ given that the epoch (in this case λ) occurs. Thus

$$\bar{\mu}_S(\lambda, \sigma) = \mu_{s_1}^\lambda \pi_{s_2}^\sigma. \quad (7.28)$$

This completes the construction of the auto-events $\bar{\xi}_S^\lambda$, $\lambda \in \Lambda''_1 \times \Sigma_{p_2}$.

The remaining elements of $\bar{\Xi}$ —those $\bar{\xi}_S^\lambda$ for $\lambda \in \Lambda''_2 \times \Sigma_{p_1}$ —are found by repeating the procedure with the roles of p_1 and p_2 reversed.

7.3 CONSOLIDATED EXOGENOUS EVENTS

The exo-event sets of elements e_1 and e_2 are the sets

$$Z_1(p) = \{\xi^\sigma : \sigma \in \Sigma_p\} \quad p \in P_1 \quad (7.29)$$

$$Z_2(p) = \{\xi^\sigma : \sigma \in \Sigma_p\} \quad p \in P_2 \quad (7.30)$$

respectively, while that of \bar{e} is

$$\bar{Z}(p) = \{\bar{\xi}^\sigma : \sigma \in \Sigma_p\} \quad p \in \bar{P}. \quad (7.31)$$

Since the event sets Z_1 and Z_2 represent the consequences of epochs determined externally to their respective elements but identified with a port, they must continue to be described in \bar{Z} . We have already seen how the exo-events corresponding to Σ_{p_1} and Σ_{p_2} are accounted for—they are absorbed into auto-events. On the other hand, if an exo-event produces an endo-epoch class on p_1 it is clear that ξ^σ , $\sigma \in \Sigma_{p_2}$, must influence the exo-event in much the same manner as for auto-events. In fact, the consolidation of exo-events follows that of auto-events almost exactly.

Figure 7.8 illustrates some typical situations for epochs observed at a particular port, say p_a . In case (a), the epoch $\sigma_1 \in \Sigma_{p_a}$ generates an endo-event at p_1 , which in turn excites the external epoch $\sigma_2 \in \Sigma_{p_2}$. However, as seen in case (b), it is possible that an external epoch σ_1 can fail to involve σ_2 . It is also possible that more than one external epoch can be randomly excited by σ_1 as suggested by (a) and (c).

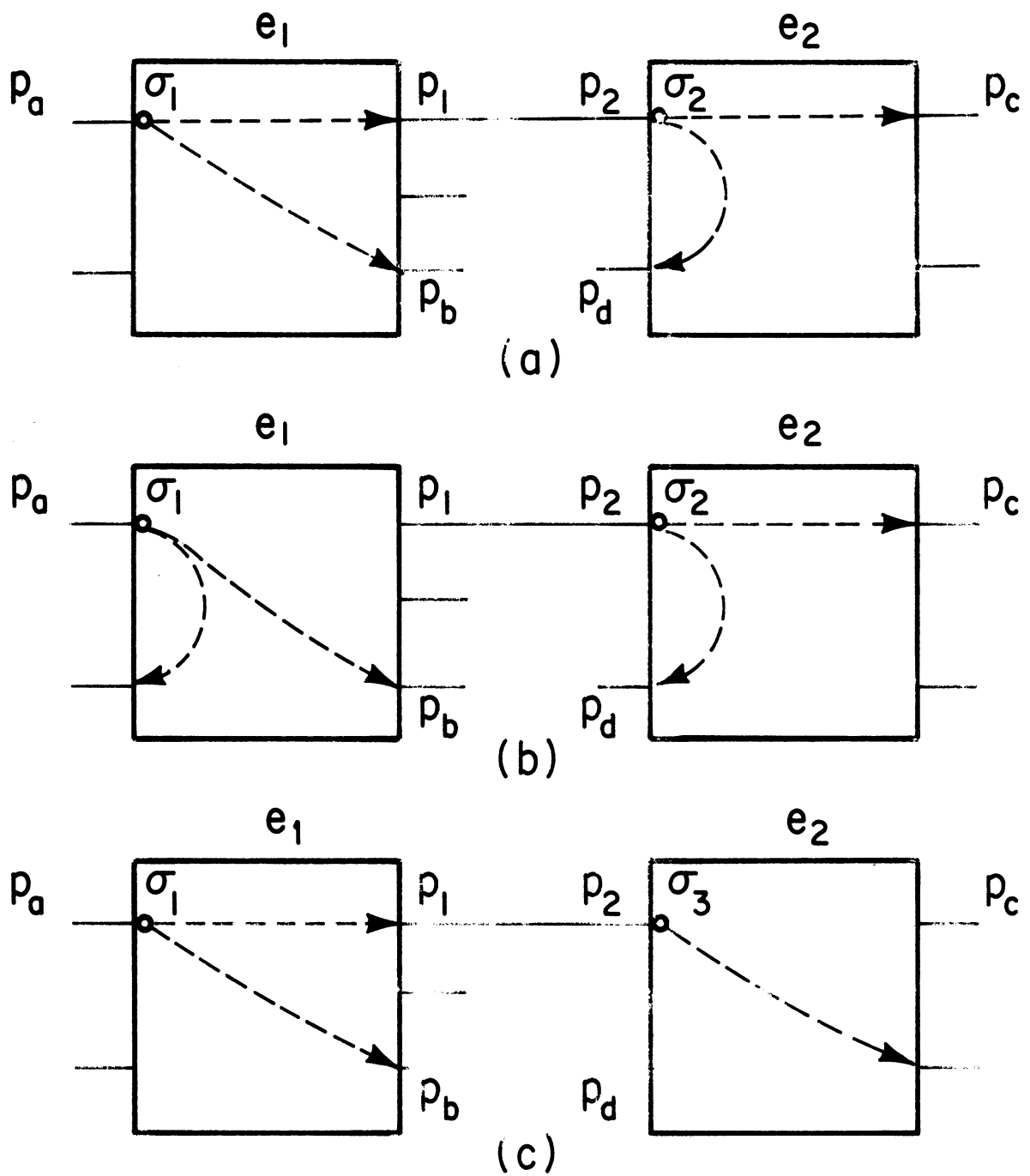


Figure 7.8

Illustrating externally determined epochs and their influence.

We recall that every exo-event of an element e is of the form

$$\xi_s^\sigma = (Q^\sigma, \alpha_s^\sigma, \beta_s^\sigma, g_s^\sigma, \pi_s^\sigma), \quad \sigma \in \Sigma_p, p \in P_e, s \in S_e, \quad (7.32)$$

and that Q^σ represents the ports excited when the external epoch σ is excited (at port p). We again distinguish two useful subsets of every Σ_p for every $p \in (P_1 - \{p_1\})$:

(1) Those exo-epochs σ for which $p_1 \notin Q^\sigma$. Call these Σ'_p

$$\Sigma'_p = \{\sigma: \sigma \in \Sigma_p, p_1 \notin Q^\sigma\}. \quad (7.33)$$

(2) Those exo-epochs σ for which $p_1 \in Q^\sigma$. Call these Σ''_p .

$$\Sigma''_p = \{\sigma: \sigma \in \Sigma_p, p_1 \in Q^\sigma\}. \quad (7.34)$$

A similar partitioning for $p \in P_2$ can be found by replacing P_1 by P_2 , p_1 by p_2 , and S_1 by S_2 in the above

Case (1)

Let

$$\bar{\xi}_s^\sigma = (\bar{Q}^\sigma, \bar{\alpha}_s^\sigma, \bar{\beta}_s^\sigma, \bar{g}_s^\sigma, \bar{\pi}_s^\sigma) \quad (7.35)$$

for all $s \in \bar{S}$ and $\sigma \in \bar{\Sigma}_p$, $p \in \bar{P}$. Let Σ'_p be contained in $\bar{\Sigma}_p$, and let

$$\bar{Q}^\sigma = Q^\sigma \quad (7.36)$$

$$\bar{\alpha}_{(s_1, s_2)}^\sigma = \alpha_{s_1}^\sigma \quad (7.37)$$

$$\bar{\beta}_{(s_1, s_2)}^\sigma = \beta_{s_1}^\sigma \quad (7.38)$$

$$\bar{g}_{(s_1, s_2)}^\sigma = (g_{s_1, s_2}^\sigma) \quad (7.39)$$

and

$$\bar{\pi}^{\sigma}(s_1, s_2) = \pi^{\sigma}_{s_1} \quad (7.40)$$

for all $(s_1, s_2) \in \bar{S}$, $\sigma \in \Sigma'_p$, $p \in P_1 - \{p_1\}$. Then $\bar{\xi}^{\sigma}_s$ are the exo-events of \bar{e} corresponding to the $\sigma \in \Sigma'_p$, $p \in P_1 - \{p_1\}$. The $\bar{\xi}^{\sigma}_s$ for $\sigma \in \Sigma'_p$, $p \in P_2 - \{p_2\}$ can be found analogously.

Case (2)

In this case, there will be an exo-event of \bar{e} for each element $\Sigma''_p \times \Sigma_{p_2}$, $p \in P_1 - \{p_1\}$; and analogously for each element of $\Sigma'_p \times \Sigma_{p_1}$, $p \in P_2 - \{p_2\}$. Consequently, we let $\bar{\Sigma}_p \triangleq \Sigma'_p \cup (\Sigma''_p \times \Sigma_{p_2})$ for $p \in P_1 - \{p_1\}$ and $\bar{\Sigma}_p \triangleq \Sigma'_p \cup (\Sigma'_p \times \Sigma_{p_1})$ for $p \in P_2 - \{p_2\}$. The exo-event set

$$\bar{Z}(p) = \{\bar{\xi}^{\sigma}: \sigma \in \bar{\Sigma}_p\}, \quad p \in \bar{P} = P_1 \cup P_2 - \{p_1, p_2\}, \quad (7.41)$$

will be a complete set of exo-events for \bar{e} .

The $\bar{\xi}^{\sigma}$ for $\sigma \in \Sigma'_p \times \Sigma_{p_2}$, $p \in P_1 - \{p_1\}$, are defined very analogously to the way the $\bar{\xi}^{\lambda}$ were defined. For example, the endo-epoch set \bar{Q}^{σ} is given by

$$\bar{Q}^{(\sigma_1, \sigma_2)} = (Q^{\sigma_1} - \{p_1\}) \cup Q^{\sigma_2} \quad (7.42)$$

The basic sequence for the formation of the endocondition functions, endocontrol functions, target state functions, and conditional probabilities for this and the "reversed roles" case will not be detailed here. It is notationally tedious, and will add little to understanding which the

previous cases have not already provided.

7.4 AN EXAMPLE OF CONSOLIDATION

Element consolidation $\mathcal{C}'(e_1, e_2, p_1, p_2)$ will be performed to illustrate the technique. In this case e_1 will be a queue, and e_2 a server. The output of the former is connected to the input of the latter, and it is this connection which is to be absorbed (see Figure 7.9).

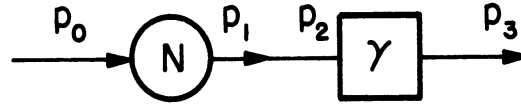


Figure 7.9 Queue-and-server consolidation.

From Table III we take the following data, with a slight change in notation. (Every equation in this section is assumed to be valid for all $s \in \bar{S}$, $s_1 \in S_1$, $s_2 \in S_2$).

The Queue

- (1) Ports: $P_1 = \{p_0, p_1\}$
- (2) States: $S_1 = \{0, 1, \dots, N\}$
- (3) Auto-event set: $\Xi_1 = \emptyset$
- (4) Exo-event set: $Z_1(p_0) = \{\zeta_{s_1}^{\sigma_{10}}\}$
 $Z_1(p_1) = \{\zeta_{s_1}^{\sigma_{11}}\}$

(5) Exoevents:

$$(a) \quad \zeta_{s_1}^{\sigma_{10}} = (Q^{\sigma_{10}}, \alpha_{s_1}^{\sigma_{10}}, \beta_{s_1}^{\sigma_{10}}, g_{s_1}^{\sigma_{10}}, \pi_{s_1}^{\sigma_{10}}), \quad \text{where}$$

$$Q^{\sigma_{10}} = \{p_1\}$$

$$\alpha_{s_1}^{\sigma_{10}}(x_1) = N - s_1 + x_1$$

$$\beta_{s_1}^{\sigma_{10}}(x_1, y_0) = s_1 + y_0$$

$$g_{s_1}^{\sigma_{10}}(x_1, y_0) = [s_1 + y_0 - x_1]_0^N$$

$$\pi_{s_1}^{\sigma_{10}} = 1$$

$$(b) \quad \zeta_{s_1}^{\sigma_{11}} = (Q^{\sigma_{11}}, \alpha_{s_1}^{\sigma_{11}}, \beta_{s_1}^{\sigma_{11}}, g_{s_1}^{\sigma_{11}}, \pi_{s_1}^{\sigma_{11}}), \quad \text{where}$$

$$Q^{\sigma_{11}} = \{p_0\}$$

$$\alpha_{s_1}^{\sigma_{11}}(x_0) = N - s_1 + x_0$$

$$\beta_{s_1}^{\sigma_{11}}(x_0, y_1) = N - s_1 + y_1$$

$$g_{s_1}^{\sigma_{11}}(x_0, y_1) = [s_1 - y_1 + x_0]_0^N$$

$$\pi_{s_1}^{\sigma_{11}} = 1$$

The Server

(1) Ports: $P_2 = \{p_2, p_3\}$

(2) States: $S_2 = \{0, 1\}$

(3) Auto-event set: $\Xi_2 = \{\xi_{s_2}^\lambda\}$

(4) Exo-event set: $Z_2(p_2) = \{\xi_{s_2}^{\sigma_{22}}\}$

$$Z_2(p_3) = \{\xi_{s_2}^{\sigma_{23}}\}$$

(5) Auto-events: $\xi_{s_2}^\lambda = (Q^\lambda, \beta_{s_2}^\lambda, g_{s_2}^\lambda, \mu_{s_2}^\lambda)$, where

$$Q^\lambda = \{p_2, p_3\}$$

$$\beta_{s_2}^\lambda(x_2, x_3) = (I_{x_3}(J^+), 1) \triangleq (\beta_2^\lambda(x_2, x_3), \beta_3^\lambda(x_2, x_3))$$

$$g_{s_2}^\lambda(x_2, x_3) = I_{x_2}(J^+)I_{x_3}(\{0\})$$

$$\mu_{s_2}^\lambda = \gamma I_{s_2}(\{0\})$$

(6) Exo-events

(a) $\xi_{s_2}^{\sigma_{22}} = (Q^{\sigma_{22}}, \alpha_{s_2}^{\sigma_{22}}, \beta_{s_2}^{\sigma_{22}}, g_{s_2}^{\sigma_{22}}, \pi_{s_2}^{\sigma_{22}})$, where

$$Q^{\sigma_{22}} = \emptyset$$

$$\alpha_{s_2}^{\sigma_{22}} = 1 - s_2$$

$$\beta_{s_2}^{\sigma_{22}} = \emptyset$$

$$g_{s_2}^{\sigma_{22}}(y_2) = 1 - I_{y_2}(\{0\}) I_{s_2}(\{0\})$$

$$\pi_{s_2}^{\sigma_{22}} = 1$$

$$(b) \quad \zeta_{s_2}^{\sigma_{23}} = (Q^{\sigma_{23}}, \alpha_{s_2}^{\sigma_{23}}, \beta_{s_2}^{\sigma_{23}}, g_{s_2}^{\sigma_{23}}, \pi_{s_2}^{\sigma_{23}}), \text{ where}$$

$$\zeta_{s_2}^{\sigma_{23}} = \emptyset$$

$$\alpha_{s_2}^{\sigma_{23}} = 0$$

$$\beta_{s_2}^{\sigma_{23}} = \emptyset$$

$$g_{s_2}^{\sigma_{23}}(y_3) = s_2$$

$$\pi_{s_2}^{\sigma_{23}} = 1$$

The Consolidation

(1) Ports: The port set is $P_1 \cup P_2 - \{p_1 - p_2\}$, hence

$$\bar{P} = \{p_0, p_3\} \quad (7.43)$$

(2) States: The state set \bar{S} is contained in $S_1 \times S_2$ hence

$$\bar{S} \subseteq \{0, 1, \dots, N\} \times \{0, 1\} \quad (7.44)$$

(3) Auto-events: Let $\bar{\xi}_s^\lambda$, $s \in \bar{S}$, be the consolidated auto-event resulting from the autoevent $\xi_{s_2}^\lambda$ of the server. Let

$$\xi_s^\lambda = (\bar{Q}^\lambda, \bar{\beta}_s^\lambda, \bar{g}_s^\lambda, \bar{\mu}_s^\lambda). \quad (7.45)$$

Then, since $Q^\lambda = \{p_2, p_3\}$, the epoch λ gives rise to an exoevent $\xi_{s_1}^{\sigma_{11}}$ at p_1 (which is connected to p_2), and by Eq. (7.19) (with roles of p_1 and p_2 reversed).

$$\bar{Q}^\lambda = Q^\lambda \cup Q^{\sigma_{11}} - \{p_2\} = \{p_0, p_3\}. \quad (7.46)$$

To obtain the endocontrol function $\bar{\beta}_s^\lambda(x_0, x_3)$ which has two components which will be called $\bar{\beta}_0^\lambda(x_0, x_3)$ and $\bar{\beta}_3^\lambda(x_0, x_3)$, we adjoin $\beta_{s_1}^{\sigma_{11}}$ (from $\xi_{s_1}^{\sigma_{11}}$) and β_3^λ (from $\xi_{s_2}^\lambda$), leaving

$$\bar{\beta}_s^\lambda(x_0, x_3) = (\bar{\beta}_0^\lambda(x_0, x_3), \bar{\beta}_3^\lambda(x_0, x_3)) = (\beta_{s_1}^{\sigma_{11}}(x_0, y_1), \beta_3^\lambda(x_2, x_3)) = (n-s_1+y_1, 1)$$

Then we replace y_1 by the endocontrol β_2^λ (from $\xi_{s_2}^\lambda$) to get

$$\bar{\beta}_s^\lambda(x_0, x_3) = (N-s_1+I_{x_3}(J^+), 1) \quad (7.47)$$

Since this does not involve x_2 , the exocondition at port 2, no further substitutions are necessary, and $\bar{\beta}_s^\lambda$ is described for all $s = (s_1, s_2) \in \bar{S}$.

To obtain the target state $\bar{g}_s^\lambda(x_0, x_3)$ which has two components $(\bar{g}_1^\lambda(x_0, x_3), \bar{g}_2^\lambda(x_0, x_3))$ we adjoin $g_{s_2}^{\sigma_{11}}(x_0, y_1)$ and $g_{s_2}^\lambda(x_2, x_3)$, producing

$$\begin{aligned} \bar{g}_s^\lambda(x_0, x_3) &= (\bar{g}_1^\lambda(x_0, x_3), \bar{g}_2^\lambda(x_0, x_3)) = (g_{s_1}^{\sigma_{11}}(x_0, y_1), g_{s_2}^\lambda(x_2, x_3)) \\ &= ([s_1-y_1+x_0]_0^N, I_{x_2}(J^+)I_{x_3}(\{0\})). \end{aligned} \quad (7.48)$$

Replacing y_1 by $\beta_2^\lambda(x_2, x_3)$ and x_2 by $\alpha_{s_1}^{\sigma_{11}}(x_0)$, and noting that

$I_{n-s_1+x_0}(J^+) = I_{n-s_1}(J^+)I_{x_0}(J^+)$ because of the non-negativity of $N-s_1$ and x_0 , we obtain

$$\bar{g}_s^\lambda(x_0, x_3) = ([s_1 - I_{x_3}(J^+) + x_0]_0^N, I_{N-s_1}(J^+)I_{x_0}(J^+)I_{x_3}(\{0\})), \quad (7.49)$$

which is the final form for \bar{g}_s^λ .

Finally, to obtain $\bar{\mu}_s^\lambda$, we take the product of $\mu_{s_2}^\lambda$ and $\pi_{s_1}^{\sigma_{11}}$ to get

$$\bar{\mu}_s^\lambda = \gamma I_{s_2}(\{0\}) \quad (7.50)$$

$$(4) \text{ Auto-event set: } \bar{\xi} = \{\bar{\xi}_s^\lambda\} \quad (7.51)$$

$$(5) \text{ Exo-event set: } \begin{aligned} Z(p_0) &= \{\bar{\xi}_s^{\sigma_0}\} \\ Z(p_3) &= \{\bar{\xi}_s^{\sigma_3}\} \end{aligned} \quad (7.52)$$

(6) Exo-events:

(a) Let the exo-event at port 0 of the consolidated network be

$$\bar{\xi}_s^{\sigma_0} = (\bar{Q}^{\sigma_0}, \bar{\alpha}_s^{\sigma_0}, \bar{\beta}_s^{\sigma_0}, \bar{g}_s^{\sigma_0}, \bar{\pi}_s^{\sigma_0}) \quad (7.53)$$

Then, since $Q_{s_1}^{\sigma_{10}} = \{p_1\}$, an exo-epoch class at port 0 produces, in turn, an endo-epoch class at port 1 and an exo-event at port 2. Thus $\bar{\xi}_{s_1}^{\sigma_{10}}$ and $\bar{\xi}_{s_2}^{\sigma_{22}}$ are involved in this exo-event. We have

$$\bar{Q}^{\sigma_0} = Q^{\sigma_{10}} \cup Q^{\sigma_{22}} - \{p_1\} = \emptyset. \quad (7.54)$$

For the endocondition function $\bar{\alpha}_s^{\sigma_0}$ at port 0 we have

$$\bar{\alpha}_s^{\sigma_0} = \alpha_{s_1}^{\sigma_{10}}(x_1) = N - s_1 + x_1. \quad (7.55)$$

Replacing x_1 by $\alpha_{s_2}^{\sigma_{22}}$, which is $1 - s_2$, we get finally

$$\bar{\alpha}_s^{\sigma_0} = N + 1 - (s_1 + s_2). \quad (7.56)$$

Because \bar{Q}^{σ_0} is empty, the endocontrol function $\bar{\beta}_s^{\sigma_0}$ specifies the endocontrol at no ports, and hence is a null function \emptyset (has null range and domain). This would also be found through systematic procedure since $\beta_{s_2}^{\sigma_{22}} = \emptyset$ also. Thus

$$\bar{\beta}_s^{\sigma_0} = \emptyset. \quad (7.57)$$

The target state $\bar{g}_s^{\sigma_0}(y_0)$ has two components, and we let $\bar{g}_s^{\sigma_0}(y_0) = (\bar{g}_1^{\sigma_0}(y_0), \bar{g}_2^{\sigma_0}(y_0))$. These are found through the substitutions

$$\begin{aligned} \bar{g}_s^{\sigma_0}(y_0) &= (\bar{g}_1^{\sigma_0}(y_0), \bar{g}_2^{\sigma_0}(y_0)) = (g_{s_1}^{\sigma_{10}}(x_1, y_0), g_{s_2}^{\sigma_{22}}(y_2)) \\ &= ([s_1 + y_0 - x_1]_0^N, 1 - I_{y_2}(\{0\})I_{s_2}(\{0\})). \end{aligned} \quad (7.58)$$

Then replacing y_2 by $\beta_{s_1}^{\sigma_{10}}(x_1, y_0)$ and x_1 by $\alpha_{s_2}^{\sigma_{22}}$, we get

$$\begin{aligned} \bar{g}_s^{\sigma_0}(y_0) &= ([y_0 + (s_1 + s_2) - 1]_0^N, 1 - I_{s_1 + y_0}(\{0\})I_{s_2}(\{0\})) \\ &= ([y_0 + (s_1 + s_2) - 1]_0, 1 - I_{s_1}(\{0\})I_{y_0}(\{0\})I_{s_2}(\{0\})). \end{aligned} \quad (7.59)$$

Finally, $\bar{\pi}_s^{\sigma_0} = \pi_{s_1}^{\sigma_{10}} \pi_{s_2}^{\sigma_{22}}$, or

$$\bar{\pi}_s^{\sigma_0} = 1 \quad (7.60)$$

(b) Let the exo-event at port 3 of the consolidated network be

$$\bar{\zeta}_s^{\sigma_3} = (\bar{Q}^{\sigma_3}, \bar{\alpha}_s^{\sigma_3}, \bar{\beta}_s^{\sigma_3}, \bar{g}_s^{\sigma_3}, \bar{\pi}_s^{\sigma_3}) \quad (7.61)$$

Then, since $\bar{Q}_{s_2}^{\sigma_{23}} = \emptyset$, an exo-event at port 3 produces no endo-events at any other ports, and only $\bar{\zeta}_{s_3}^{\sigma_{23}}$ is involved in this event (i. e. this is a case (1) consolidation; see Eqs. 7.36 through 7.41). Consequently

$$\bar{Q}^{\sigma_3} = \emptyset \quad (7.62)$$

$$\bar{\alpha}_s^{\sigma_3} = \alpha_{s_2}^{\sigma_{23}} = 0 \quad (7.63)$$

$$\bar{\beta}_s^{\sigma_3} = \emptyset \quad (7.64)$$

$$\bar{g}_s^{\sigma_3}(y_3) = g_s^{\sigma_{23}}(y_3) = s_2 \quad (7.65)$$

$$\bar{\pi}_s^{\sigma_3} = \pi_s^{\sigma_{23}} = 1 \quad (7.66)$$

This completes the consolidation. A careful examination of each of the properties of this element reveals that its description is exactly the description that would be expected. This definition is the representation of a new element, which could be added to our list of primitive elements (Table I).

7.5 CONSOLIDATED STATE SET

The set \bar{S} has not been defined, except to say that it is contained in the Cartesian product of S_1 and S_2 , $S_1 \times S_2$. The various functions representing target states, endoconditions, and endocontrols were said to be functions on this set. As a matter of fact, however, they have all been well defined over the larger state $S_1 \times S_2$. Nevertheless, $S_1 \times S_2$ is generally much larger than necessary, since there are many states in it which the process would never return to once leaving. In fact, a great many states are left as soon as any epoch occurs within their element.

A state s is said to be essential if a return is possible from every state to which s leads (see Ref. 1). Since the equilibrium probability of an inessential state is zero, little is lost in omitting it from the model. However, unrecognized inessential states can waste memory in a computation. Moreover, in typical RQA-1 models, the ratio of inessential to essential states when a Cartesian product space is used is often ten to one, and could be considerably greater.

A foolproof scheme to eliminate all inessential states with reasonable computation is very desirable, but difficult to produce. However, techniques which are very effective, but not perfect, can be devised. Mostly they involve deriving constraint equations from the target state functions. This problem will be treated in a later work.

7.6 REDUCTION OF A NETWORK TO A TRANSITION INTENSITY MATRIX

With the establishment of a well-defined consolidation operator, the problem of deriving the transition intensity matrix from a given finite network has been solved. For, by successively applying the consolidation operator to each connection in the network, one can ultimately reduce it to an equivalent network which has no ports and only a single element. If it has no ports, then this element also has no exogenous events, and the autogenous events are extremely simple. There are no endocontrols, endoconditions, or endo-events.

We observe that $N^* = (\{e^*\}, \emptyset)$ is the limit network, where \emptyset represents the null (connection) function. The element e^* must have the form

$$e^* = (S, \emptyset, \Xi, \emptyset) \quad (7.67)$$

where

$$\Xi = \{\xi^\lambda : \lambda \in \Lambda\} \quad (7.68)$$

and

$$\xi_s^\lambda = (\emptyset, \emptyset, g_s^\lambda, \mu_s^\lambda). \quad (7.69)$$

Notice that μ_s^λ is simply the probability intensity of a primitive epoch and that g_s^λ is simply a transition function

$$g_s^\lambda : S \rightarrow S. \quad (7.70)$$

Thus Ξ can be looked upon as a collection of matrix elements, so that if

we map S onto a set of integers, and let $j = g_i^\lambda$, then μ_i^λ is a probability intensity that a jump will occur from state i to state j .

This is clearly just another form for the transition intensity matrix.

7.7 NOTES

We have succeeded in obtaining what amounts to a definition of a queueing network. That this is a useful definition, one which adequately models real-world queueing systems, can be argued from experience and the rather intuitive way the model has been built up from the "kinds of things we want to talk about." This report has attempted to be formal in its notation and definitions, but not in its arguments. To have attempted to pose formal arguments at this point in the development would have significantly delayed publication (perhaps indefinitely), and thus delayed the progress that inevitably comes from the interchange of ideas with others.

Nevertheless, it is believed that an axiomatic structure, and an orderly theory can eventually be built around the arguments and results presented here. It is further believed that the notions developed can be readily extended to networks which are not tree-structured, and to networks which must be modeled by semi-Markov processes rather than Markov chains.

References

- [1] Chung, K. L., Markov Chains with Stationary Transition Probabilities, Springer Verlag, Hamburg, 1960.
- [2] Wallace, V. L. and R. S. Rosenberg, "RAQ-1, The Recursive Queue Analyzer," Systems Engineering Laboratory Technical Report no. 2, Electrical Engineering Department, University of Michigan, February 1966.
- [3] Sutherland, W. R., "The On-Line Graphical Simplification of Computer Procedures," Ph.D. thesis, Electrical Engineering Department, Massachusetts Institute of Technology, January 1966.
- [4] Gordon, G., "A General Purpose Systems Simulation Program," Proc. Eastern Joint Computer Conference, Washington, D.C., December 1961.
- [5] Morse, P. M., Queues Inventories and Maintenance, John Wiley and Sons, New York, 1958.
- [6] Irani, K. B. and V. L. Wallace, "A System for the Solution of Simple Stochastic Networks," Technical Report 31, Systems Engineering Laboratory, Department of Electrical Engineering, University of Michigan, Ann Arbor, 1969; also Technical Report no. 14, Concomp Project, Computing Center, University of Michigan, Ann Arbor, 1969.
- [7] Randall, L. S., I. S. Uppal, G. A. McClain, and J. F. Blinn, "An Implementation of the Queue Analyzer System (QAS) on the IBM 360/67," Technical Report 43, Systems Engineering Laboratory, Department of Electrical Engineering, University of Michigan, Ann Arbor, 1969; also Technical Report 22, Concomp Project, Computing Center, University of Michigan, Ann Arbor, 1969.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be at least as high as the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
CONCOMP PROJECT UNIVERSITY OF MICHIGAN		2b. GROUP	
3. REPORT TITLE			
ON THE REPRESENTATION OF MARKOVIAN SYSTEMS BY NETWORK MODELS			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
Technical Report 21			
5. AUTHOR(S) (First name, middle initial, last name)			
VICTOR L. WALLACE			
6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS	
August 1969	112	7	
8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)		
DA-49-083 OSA-3050	Concomp Technical Report 21		
b. PROJECT NO.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)		
c.	SEL TR 42		
10. DISTRIBUTION STATEMENT			
Qualified requesters may obtain copies of this report from DDC.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Advanced Research Projects Agency	
13. ABSTRACT			
<p>Formal, unambiguous mathematical structures are developed for representing Markovian queueing networks and for systematically constructing a description of a continuous-parameter Markov chain model from a description of the network diagram. A formal queueing diagram notation is developed as a pictorial language. An approach to the problem decomposition and recomposition of Markovian queueing networks is presented, and applied to realistic queueing networks.</p>			

14.	KEY WORDS	LINK A		LINK B		LINK C	
		ROLE	WT	ROLE	WT	ROLE	WT
	pictorial language Markovian queueing networks networks mathematical modeling						