DISTRIBUTED HIERARCHICAL AUTOMATA

WITH APPLICATIONS TO GENETICS IN PROCARYOTES

by

Wolf Kohn

B.A., Universidad Nacional de Colombia, Bogota
(1970)

S.M., Massachusetts Institute of Technology
(1974)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1978

Signature of Author.....
Department of Electrical Engineering
and Computer Science, January 13, 1978

Certified by.....
Thesis Supervisor

Accepted by.....
Chairman, Departmental Committee on Graduate Students

DISTRIBUTED HIERARCHICAL AUTOMATA

WITH APPLICATIONS TO GENETICS IN PROCARYOTES

by

Wolf Kohn

Submitted to the Department of Electrical Engineering & Computer Science
on January 13, 1978 in partial fulfillment of the requirement
for the Degree of Doctor of Philosophy

ABSTRACT

This study is concerned with the development and analysis of a formal model for a computing device composed of two-level finite state interacting elements allocated in an unbounded K-dimensional grid: the distributed hierarchical automata. The model is used as the main tool for the study of three types of epigenetic control processes in the genetic apparatus of procaryotes: Initial Activation Processes, Composite Processes, and Construction Processes; and also, for the development of a structural identification algorithm for determining dynamic characteristics of epigenetic processes in the basis of observed experimental data.

Thesis Supervisor: Alan S. Willsky
Title: Associate Professor of Electrical Engineering & Computer Science

DEDICATED TO MY PARENTS

Salomon and Hilda Kohn

ACKNOWLEDGEMENTS

The research carried out in this thesis could not have been accomplished without the support and guidance of Professor Alan S. Willsky. I sincerely acknowledge his patient, and sometimes trying efforts during the preparation of this thesis.

I would like to thank Professor Sanjoy K. Mitter, not only for his knowledgeable advice during the preparation of this thesis, but also for his friendship during my stance at M.I.T. The interaction with Prof. Mitter, not only as his student and Graduate T.A., but as a friendly ear, is one of the most profitable events of my academic experience.

Professor Timothy L. Johnson, who first introduced me to the wonderful world of biological systems, had a definite influence on my education, not only during the research of this thesis, but also throughout my collaboration with him in several biologically-related projects at the Electronic Systems Laboratory.

I want to thank Dr. Gail Willsky, of Harvard University, for her helpful advice during my research, and most of all for agreeing to discuss genetics in a systems' context and at my level of understanding. The outcomes of these discussions are reflected, in part, in the organization and context of Chapters 2 and 3 of this document.

I would like to thank Professor Michael Arbib, of U. Mass. at Amherst, for his helpful comments and discussions on several aspects of distributed automata, some of which were considered in this thesis.

During my stance at M.I.T., my (unconscious) tendency of defeating any form of administrative order was, fortunately, not successful. I thank, for that, two friends; Mr. Richard A. Osborne at the Electronic Systems Laboratory and Ms. Marilyn Pierce of the E.E. Graduate Office.

I also want to acknowledge the profitable interaction with my fellow students at E.S.L. In particular, my friends Poh Kam Wong, Wing Hong Lee, Edward Chow and Demosthenes Teneketzis of the "night-shift" which constituted a forum in which many problems of the world were solved; and my friends of the "day-shift" Stephen Marcus, Jack Liu, Robert Washburn and Steve Young for a friendly ear to many, sometimes not totally coherent, ideas in system theory.

I want to thank Professor Leonard A. Gould, my academic counselor, for his friendship and for his advice during my years at M.I.T.

Finally, I want to thank all my friends at E.S.L., in particular, Barbara Peacock who typed several versions of this thesis and who matched my procrastinating index with hers, so that within the two of us a resonant delay was accomplished; to Arthur Giordani for a beautiful set of figures and many cigarette-chatting moments; to Maria Monserrate for a daily ration of Spanish; and, to all the others which space does not allow me to list here.

I would like to thank Susan for many things, in particular, for adapting herself to an unusual time schedule.

Last, but not least, I want to thank my parents, to whom this thesis is dedicated, for their unrestrictive moral support and many other things.

The research for this thesis was conducted at the M.I.T. Electronic Systems Laboratory with partial support provided by NSF under Grants ENG77-0777 and ENG 76-02860.

TABLE OF CONTENTS

CHAPTER		PAGE
1.	INTRODUCTION	8
	1.1 Outline of the Study	8
	1.2 Fundamental Characteristics of the Model	10
	1.3 Contributions of the Study	22
2.	GENERAL STATEMENT OF THE PROBLEM	27
	2.1 Outline of the Epigenetic Control Problem in Procaryotes	27
	2.2 Description of the Concept of Distributed Dynamics and Neighboring Interaction	59
	2.3 Scope and Aims of the Study, Fundamental Assumptions and Experimental Data Base	66
	2.4 Summary	83
3.	BASE MODEL FOR EPIGENETIC CONTROL MECHANISMS IN PROCARYOTES	85
	3.1 Outline of Modeling Strategy	85
	3.2 Classification of Elements of the Model According to Function as Information Carriers	103
	3.3 Incidence Diagram of Dynamic Interaction Among the Elements of the Classes	110
	3.4 State Aggregation of the Base Model	216
	3.5 Examples	226
	3.5.1 The Lac Operon	227
	3.5.2 A 3-Operon Eigen Cycle	256

CHAPTER		PAGE
4.	FORMAL MODEL FOR EPIGENETIC CONTROL MECHANISMS: THE DISTRIBUTED HIERARCHICAL AUTOMATON	262
	4.1 Formulation of the Model	262
	4.2 Basic Characteristics of DHA's	280
	4.2.1 Local Properties of DHA's	280
	4.2.2 The Concepts of Behavior and Display	306
	4.3 Realization Theory of DHA's	309
	4.4 Algorithmic Computation of a Covering DHA for a Given Finite Family of DHA's	331
	4.5 Process Control in DHA's	344
5.	SIMULATION STUDY OF EPIGENETIC CONTROL MECHANISMS	399
	5.1 Computer Implementation of DHA's	399
	5.2 Structural Identification of Epigenetic Processes in the Context of DHA's	429
6.	SUMMARY, CONCLUSIONS AND FUTURE RESEARCH	449
	6.1 Summary	449
	6.2 Conclusions	451
	6.3 Critique and Future Research	453
BIBLIOGRAPHY		456
APPENDIX A.1		
APPENDIX A.2		
APPENDIX A.3		

1. INTRODUCTION

1.1 Outline of the Study

The present study is concerned with the development and analysis of a formal model that represents the information-dynamics mechanisms of the genetic apparatus in procaryotic cells; that is, the mechanisms that accompany the processes by which elements (chemical entities such as proteins, metabolites, metallic ions, etc.) in the cell experience structural transformations in a coordinated fashion.

We will concentrate our effort on the information-based mechanisms in the genetic apparatus known as epigenetic control mechanisms. We briefly describe these mechanisms next.

By epigenetic control mechanisms we mean the collection of rules (based on physical and organizational principles) that govern the behavior of the cell as a biological system. Although this definition is by no means universally accepted, (see for instance, Rosen [1]) it conveys at an admittedly too general level, for the moment, the scope of our study. From those rules, we will extract a classification of the elements of the genetic apparatus according to their role in the controlled behavior of the cell, and a set of "canonical" dynamic interactions among elements of the different classes of this classification. The main criterion in the classification is to abstract those properties that determine the behavior of elements as information carriers in the genetic activity of the cell; and on the basis of this classification,

and the associated canonical interactions, develop a model in terms of which some aspects of regulation and modulation in the cell, directly dependent on the genetic structure, can be analyzed. In particular, we are interested in the following:

- controlled protein synthesis and repression processes;
- catalysis and inhibition in chemical reactions that occur in the intra-cellular space;
- genetically controlled response to environmental changes.

In principle, all the processes in the cell are determined by the structural characteristics of the elements that participate in them (in fact, this is one of the fundamental assumption in theory of developmental biology; see Wolpert [2]). However, for many studies, at the cellular level, it is sufficient to abstract some properties of the dynamics of the elements and model the elements (accordingly) as items possessing those properties without considering their physical structure. This strategy is followed in the study of epigenetic control mechanisms outlined above.

We conclude this outline by mentioning that, as in any study of physical phenomena by means of mathematical models, we have made certain idealizations concerning the physical characteristics of the processes we are modeling. We will explicitly state these whenever they arise in our study and evaluate their effects (restrictions) on conclusions that are derived from the study of our model.

In addition, in the development of the model, we have used certain

assumptions about the physics of the processes under study, that are still the subject of controversy. Whenever these arise, we will outline the experimental evidence on which they are based, and attempt to qualify their significance in our model.

Occasionally, we will use the model to test assumptions about the system it represents, or at least to predict the consequences of these assumptions and/or to aid in suggesting experiments by which the validity of these assumptions can be tested.

We close this introduction by noting that this research should be considered as an initial step towards building a foundation and developing research aids (i.e., models) on which further study in the field can be based.

1.2 Fundamental Characteristics of the Model

The methodology that we will follow in our study of epigenetic problems, is based on the development of a model that captures some important features of the genetic system of procaryotes. In a somewhat abstracted setting, some of the principal features of our model of epigenetic control processes in procaryotes are as follows:

1. There is no central controller that regulates the dynamics of the system and consequently, the flow of information is carried out via the decentralized interaction among elements of the system.
2. Coordination of activities in the system is implicit in the rules of operation of its elements.

3. The dynamics of the system is determined by the dynamics of each of its elements. In turn, the dynamic behavior of every element in the system depends only on the dynamics of a finite number of elements physically close to it in the informational space. That is, there are not actions-at-a-distance among elements of the system.

4. All the control actions required for the proper operation of the system are generated within the system and carried out by interaction among its elements.

Equivalently, the system is hierarchically closed [5]; that is, all the control actions, such as response to environmental changes (different metabolites), immunological responses, etc., are generated within the system itself. This characteristic, is, from the control point of view, the main distinctive feature between procaryotes and eucaryotes ([3]).

Now we list and sketch briefly some of the distinctive features of the elements that form the building blocks of our system; as we shall see, these features plus the system properties listed above, constitute the basic criterion for the development of the formal model. These features have been established by abstracting observed physical and/or operational traits of the elements of the system.

1. The number of elements composing the system is finite and remains finite during the life span of the system, but this number is arbitrarily large.

2. Each element can exhibit only a finite number of informational

states[1]; in particular, every element can exhibit a distinguished informational state whose information is "no information is carried by the element at this time".

3. Only finitely many of the elements of the system are in non-dormant state at every time.

4. At any time, the evolution of the state of an element depends solely on the state of neighboring elements, where for the moment, neighboring elements means a set of elements that are physically close to the element in question. We will make this concept more precise later on (see Sections 3.1 and 4.1).

5. The elements are allocated in a coordinated space that we will call the informational space, with exactly one element per coordinate, and all the elements have null-volumetric extension in this space (i.e., the elements are simply points in the space).

6. The information content of the state of any element in the system, at all times can be decomposed into two items: structure and intensity. The structure part of the state contains information about the dynamic status of the element at this time and its identity (i.e., the type of element according to the classification mentioned above). The intensity carries

[1]The concepts of information dynamics, informational state, etc. used in this outline are given in an intuitive fashion. Their formal meaning will be discussed in terms of the model in Chapter 3.

information about the element, concerned with its function in the system (e.g., an element representing a catalytic protein will carry in its intensity component of the state, information about its catalytic condition; for instance, which reactants are catalyzed by the protein (see Section 3.3). Any type of element in the system can exhibit only a finite number of different structures and intensities in its state.

The state decomposition mentioned in the last paragraph is one of the cornerstones in our criteria for model development. Its significance will be emphasized in many parts throughout this study. See Sections 2.1, 3.1 and 4.1 for examples.

7. Finally, we will assume that the system, and consequently its elements, are located in an unbounded space with a coordinate system as mentioned before, and that the availability of elements for allocation in such a space is arbitrarily large.

The above list of traits is included in order to indicate the unifying modeling philosophy to be used as we develop models for each system element.

Based on the previous informal description of the important attributes of the system, we now outline the model for this study whose characteristics are determined on the basis of those attributes. We will see in Chapter 3 that these attributes correspond to physical or operational characteristics of the elements of the genetic apparatus.

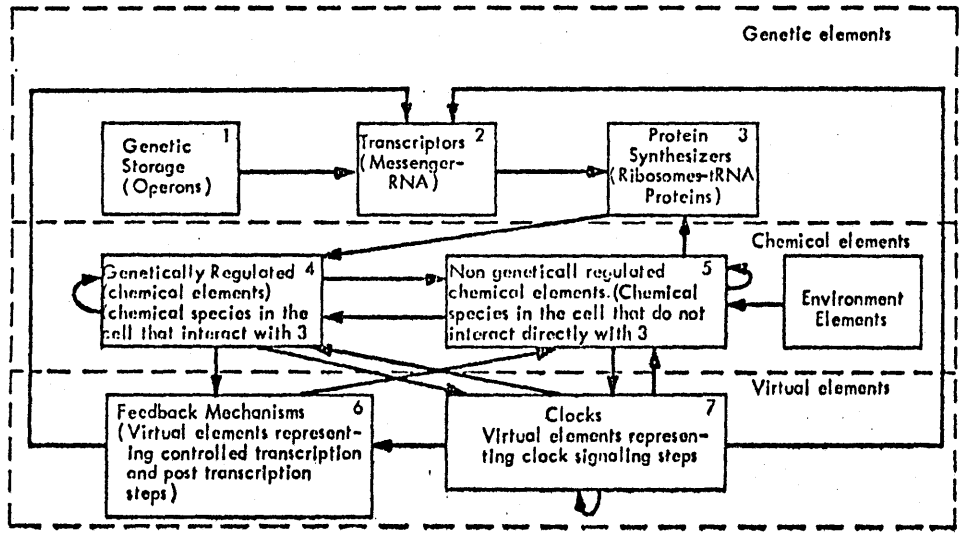


Fig. 1 Elements of the Epigenetic Control Model

Our model is composed of 3 classes of elements:

1. Genetic elements

2. Chemical elements

3. Virtual elements.

The genetic elements are further classified into 3 categories:

a) Genetic storage - which as its name indicates, represents the DNA of the cell. We assume that the information contained in the DNA is organized in indivisible units called operons ([6]), which precisely represent the information contained in the operons of the cell. Whenever an element represents a gene that is "on" we will assume that the whole operon to which this gene belongs is active. Since operons are constituted by sets of linked (although not necessarily adjacent) genes, we believe that for the purposes of this initial operational study, this assumption is adequate (see Lewin [6] for some justification from the biological point of view for this assumption).

b) Transcriptors - this category consists of polycistronic M-RNA's. They are simply copies of the operons of the genetic storage category. For our purposes, transcriptor elements are included in the model because the most important control action of the genetic apparatus is carried out at the transcriptional level [7], (i.e., repression and induction).

c) Protein synthesizers - these elements are a representation of the elements in the cell responsible for, or that participate in, the synthesis of proteins. These include ribosomes, tRNA's with and without

the respective amino acid attached to it, and regulation proteins. (In this study we will not consider structural proteins.) The model for the

The chemical elements class is also divided into 3 categories:

a) Genetically regulated chemical elements - these elements represent chemical components that interact (receive information) with the protein-synthesizer category. In other words, elements whose dynamics are regulated and/or modulated by regulation proteins. An example of these elements are the chemical elements of the lac-operon [41].

b) Non-genetically regulated chemical elements - in this category we include elements which participate in processes that are not regulated by the synthesis of proteins, that is, triggered as a response to particular states of the system. In short, the elements of this category represent chemical components which participate in chemical reactions which are not caralyzed (or modulated or inhibited) by regulation proteins. Notice that this does not imply that those reactions are not catalyzed by proteins. In fact, most of them are catalyzed by proteins, but these proteins are synthesized constitutively (Watson [8]); that is, irrespective of whether or not the reactants (or products) of the reactions they catalyze are present in the system. We note that from the biochemical dynamics point of view, the elements of this category are

analogous to the ones classified in a); however, from the information dynamics point of view they are quite different. For instance, regulation of chemical processes by competitive or constitutive inhibition does not involve responsive synthesis of proteins, but rather the control action is implicit in the interaction of the elements involved in the process. Another example of a control mechanism involving elements of this category is allostery. By allostery, we mean the control of a chemical process in the genetic apparatus by another chemical process whose elements do not directly have an affinity for the elements of the process to be regulated. The actual control is achieved via one or more chemical elements (called allosteric elements) which do possess affinity for elements in both processes (see Jacob and Monod [31]). In short, allostery is a form of indirect control. We will consider several aspects of allosteric epigenetic control in our study, in terms of the formal model.

c) Environmental elements - this is a third category of the class of chemical elements. In this category we include all the elements that enter the cellular space from the environment. As mentioned before, these include metabolites (which belong to the class of chemical elements) and phages which have genetic apparatus or rather genetic storage and use the synthesis elements of the host for synthesizing proteins.

The two classes of elements of the model discussed above contain elements that are abstractions of physical components actually present in the system (the genetic apparatus). In the third class of elements,

we do not represent physical elements of the system, but rather some very important control actions that are characteristic of the genetic apparatus; hence, the name virtual element class.

The class of virtual elements is divided into two categories:

a) Feedback mechanisms - as its name indicates, these elements represent feedback actions in the genetic apparatus. Since these feedback actions often involve processes with elements physically located in non-neighboring spots in the system and yet, the dynamics of these elements (as can be seen experimentally) is linked. We include virtual elements in our model to represent this linkage (interaction) while preserving the neighboring interaction criterion discussed previously.

However, the objectives of this classification go far beyond a simple modeling criterion. We hope that the analysis of the dynamics of these virtual elements, in terms of the model, will shed some light into the operation of non-neighboring element interactions, a study that is the object of current study in genetic research (see Ptäshne, et al [9]).

b) Clocks - these elements represent time-coordination actions in the system. Since procaryotes are, from the genetic point of view, autonomous systems, the timing of the different processes must be generated within the system itself. This timing includes a hierarchy in the dynamics of processes, which, as we shall show in the present study, is responsible for the coordinated activity of the system despite the absence of a central controller. In particular, we are interested in incorporating into our model elements that represent the mechanisms for

the coordination of the utilization of elements whose availability in the system is limited (this type of time coordination is studied in connection with many other different types of systems (see for instance Gallager, et al [10] under the heading Dynamic-conflict free-resource allocation). The study of biological clocks has occupied a central position in theoretical biology research for many years (Prigogine [11]). Nevertheless, many of their basic characteristics (from the operational and physical point of view) are poorly understood. It is our purpose in this study to find and analyze some of these characteristics in connection with the role of clocks in epigenetic control problems.

In the previous paragraphs we have given an outline of the classification of the elements of the system as a preliminary step in the modeling process. The next step that we follow is the determination of the so-called diagram of incidence of the different categories of elements in the system [12].

The diagram of incidence for our model is a flow chart in which the boxes represent categories of elements and the arrows represent information flow (see Figure 1). This diagram establishes the basis for the formulation of interaction relation among elements in the system. It also establishes the criterion for model aggregation that will be used in this study.

By model aggregation we mean a procedure for determining the appropriate dynamic characteristics of a typical element of each of the classes of elements described previously. The classification of elements

discussed above is based on the function of these elements in the system as information carriers. Thus, although two elements of the same category have very different biochemical characteristics (for instance, two different proteins of the Protein synthesizers category catalyze different reactions) their behavior as information carriers is similar, and therefore they can be modeled with elements that have the same dynamic structure from the information point of view, with the only possible difference between them being the number of different states that each one of them can attain.

The last paragraph outlines the modeling strategy we will follow for representing elements of the system:

1. For each category we develop a finite number of typical elements (from the information-dynamics point of view). The structure of any element in a category is identical to one and only one of the typical elements of that category. We will refer to the typical elements of a category as the types of that category.
2. The dynamic-interaction characteristics among types, are constrained by the information flow chart among elements of different categories (or among elements of the same category) as indicated in Figure 1. No other interactions are allowed.

The next step in the modeling process, is to determine the dynamic structure of the types in the model; that is, the evolution of their state in time. Clearly, since our model represents a physical system,

we want a causal representation of this evolution. Further, the evolution of the state of an element in any place in the system is determined by its interaction with elements located in its neighborhood. For these reasons we model the types (and thus, the elements) as finite-state output automata ([13]) with the states of neighboring elements as inputs. We note that the elements of the model are assumed to be allocated in a coordinated space (one element per coordinate) that we referred to previously as the information space. Thus, the neighborhood elements of a given element are those elements located in neighboring coordinates, a concept that will be made explicit in terms of the formal model. Models of this type have been studied in computer science under the name of cellular automata (Codd [22]). As it is usual in automata based models, the time variable takes discrete values (i.e., 0,1,2,...).

From the discussion in the last paragraph, it is clear that the dynamics of an element is completely specified once we define the state transition function of the element.

In our case, the state transition function of an element is a function that, given the present state of the element and the present states of its neighboring elements, computes the state one unit of time ahead.

Since the state of an element is composed of two parts (structure and intensity) the state transition function must be decomposable into two functions: one that computes the next structure part of the state given the present state and inputs, and another that computes the next intensity part of the state in terms of the present state and inputs.

This is the fundamental criterion for the construction of the transition functions for each of the types of the model.

For illustrative purposes only Figure 2 shows the state diagram, i.e., the graph of the transition function of a hypothetical type in the model. We note the distinction between structure (dashed line boxes) and intensity parts of the states.

With the construction of the transition function for each type, the so-called base model [12] is completed. In this model we have a description of its elements (in our case the types) and the way they interact dynamically, so that in principle, we can study the characteristics of the system in terms of the model. The task is termendously difficult, because in order to describe global dynamic characteristics of the system (as is required, for instance in the study of controlled metabolic chains [14]) we have to study the evolution of each element in the model. Due to the nature of our informational space (i.e., recall that it is unbounded) this is extremely difficult. For this and other reasons that will become apparent as we progress in this study, a formal (i.e., mathematical) model is developed on the basis of the base model outlined above. We call the procedure followed in the development of such a formal model a realization of the model.

1.3 Contributions of This Study

Our goal is to study some operational aspects of the information dynamics in the genetic system of procaryotes. In particular, our empha-

sis is in the control aspects of the processes that regulate and modulate those operational characteristics. We will look at these control mechanisms in the setting or an epigenetic description of the elements that participate in this process and their characteristic interaction that determines their ability to respond in coordinated fashion to different environmental conditions.

The methodology we use in our study can be summarized in the following steps:

(i) Development of a base model for these mechanisms based on an analysis of the operative characteristics of the elements that participate in these processes. We concentrate our effort on those characteristics that involve transformation or storage of genetic information (see Chapters 2 and 3).

(ii) We define a finite set of distinguishing characteristics and classify the genetic elements according to these characteristics (see Sections 3.1 - 3.4). This step involves an aggregation procedure which consists of defining, for each class of elements, a model typical of this class.

(iii) In this step, we develop a formal methematical model, based on the base model, which preserves most of the important operative characteristics of the base model such as the fact that the dynamics of processes in the genetic apparatus are distributed in the cellular space, the control of these processes is not the responsibility of a central control entity, and the ele-

ments oarticipating in these processes share a common hierarchical structure.

This model is the distributed hierarchical automaton (DHA) whose main features are described in Chapter 4.

The DHA model we have developed seems to be new and we show it possesses properties that allow its analysis and construction with effective i.e., algorithmic procedures (see Sections 3.5 and 5.1). In particular, the realization problem of DHA is carried out with an algebraic procedure that is a generalization of the corresponding procedure for finite state machines. To our knowledge, this is the first time such an approach has been attempted for distributed automata. Similarly, the determination of the universal type; that is, the simulation of a DHA by another DHA (the covering DHA (see Section 4.4)) is carried out by an algorithmic procedure (whose objective is to obtain a single module capable of emulating any of the modules of the DHA it is simulating). Last, but not least, three kinds of control problems and their solutions, formulated on DHA's involving processes such as construction and process-global regulation are new (see Section 4.5).

(iv) This step involves the implementation of the model developed in step (iii) as a computer program (see Chapter 5). This program allows us to simulate the DHA and determine from these simulations some additional properties and its computational shortcomings as a research tool in the study of epigenetic control processes. The program developed in this step incor-

porates several novel ideas for simulation of large scale dynamic systems (such as the DHA) such as the use of hashing functions for storing and retrieving data for the simulation of the dynamics of these systems.

(v) Finally, in this step, we use the program developed in the previous step, and apply steps (i)-(iii) to a few chosen genetic processes that actually occur in the cell in order to evaluate both the performance of the model and methodology in the study of practical problems, and to determine its limitations in this respect.

We note that in the DHA for epigenetic control models we have made several assumptions which are motivated by mathematical and computational factors rather than physical factors - we mention a few - unboundedness of the information space, finiteness and quantization of the state of the types, discretization of the time scale, etc. We will use the application examples of step (v), in part, to determine how severe these approximations are with regards to the usefulness of the model as a research tool in epigenetic studies.

In synthesis, our accomplishment with this research effort was to develop a model that allows us to do in-numero studies of several genetic problems, to exhibit its usefulness as a research tool in the study of those problems, and to determine its limitations in these tasks.

We conclude this section by mentioning that the construction of the model object of this study was carried out with the fundamental goal of

providing a flexible computational tool for studying a great variety of epigenetic control processes rather than a particular process. With this objective in mind we have developed a conversational algorithm so that a user which may not be interested in the internal structure of the model, but rather in its applicability, can proceed to use it with only a very limited knowledge of how the model works, since the computation of the model proceeds from biological characteristics requested by the construction program (Section 3.5) to the user. The translation of these characteristics into operational code is automatic, and the outcome of the simulation of the model is given to the user also in biological terms, (see Sections 3.5 and 5.1).

We illustrate several characteristics of the model using as an example the lac-operon system in E. coli. This system has the advantage of being simple enough so that different aspects of the model can be illustrated without obscuring them with the technical difficulties that arise in a more complicated system. Also, this system has been widely studied so that experimental data about its behavior is readily available and we can correlate it with the corresponding behavior obtained from the model.

2. GENERAL STATEMENT OF THE PROBLEM

2.1 Outline of the Epigenetic Control Problem in Procaryotes

In this chapter we establish, in general terms, the aims and scope of our study of epigenetic control processes in procaryotes.

As a preliminary step, we introduce some concepts of theoretical genetics with emphasis on their interpretation in the context of our study. Next, we formulate the objective of our study, subdivide it into three subproblems and discuss the justification of this classification. We conclude this chapter with a brief description of the methodology used in the analysis of the problem, and a critical assessment of the advantages and limitations of our approach, as compared with other procedures and techniques that have been used in the past to study the problem. (i.e., [1], [2]).

A generally accepted fact in modern biology states that biological systems are formed by a finite set of different components, (this is sometimes referred to as the modular principle [3]) whose dynamic interaction characterizes the dynamics of the system. In particular, procaryotic cells[1] are considered as systems composed of a finite number of molecular species whose characteristics determine the dynamic behavior of the cell.

We note that molecules themselves can be thought of as dynamical

---

[1]Procaryotes are organisms composed of a single cell. (see Watson [20]).

systems whose elements are the atoms that compose them; however, in epigenetic studies it is customary to consider molecules as the basic elements of the system under study.

The previous remark establishes the first important characteristic of epigenetic studies, namely, that the building blocks of the system are molecules.

From the physical point of view, the study of dynamics of ensembles of molecules has evolved into an integrated body of techniques and principles called macroscopic or phenomenological kinetics [4]. This name is justified by the fact that it has its origins and inspiration in the interpretation of experimental data at the compositional level of molar concentrations of each of the molecules in the system, with respect to a reference volume (e.g. the intracellular volume).

The macroscopic kinetics of a system of molecules is usually represented by a set of coupled ordinary differential equations which attempt to simulate the time-evolution of the molar concentrations of the molecules in the system. Because of the relation that this model has with our model of epigenetic mechanisms, we describe it briefly.

Lets assume that the system consists of N distinct types of molecules of known initial concentration enclosed in a fixed volume (thus, the system is isometric). With each type i of molecule in the system a variable, $x_i(t)$, (t is the time variable) is associated. Here, $x_i(t)$ represents the molar concentration of the i-th

species at time t. We further assume that the system activity is characterized by M chemical reactions among the N species of the system. (We assume that both reactants and products of these reactions are part of the N species in the system).

The macrokinetic model of a system of the type described above, is derived by expressing the speed of increase or decrease in time of each of the molecular species in the system as a function of some of the molar concentrations in the system. In symbols,

$$\dot{x}_i(t) = f_i(x_1(t)\ldots x_N(t)) \quad i=1,\ldots,N \tag{1}$$

$$x_i(o) = x_i o, \text{ given}$$

where the functions f_i, $i=1,\ldots,N$ are determined by a combination of theoretical and empirical considerations such as mass balance and observed reaction courses, and most important, by <u>assuming</u> reaction mechanisms for each of the M reactions of the system. A particularly popular example of this method of study is provided by the so-called Michaelis-Menten model ([5], [6], [7]) which, although initially developed for studying enzyme-kinetics, has been generalized to represent a large variety of molecular systems.

The objectives behind the methodologies of macroscopic kinetics outlined above, is to establish physical and operational characteristics of the system in terms of the mathematical properties of its concentration-dynamics model (i.e., a set of equations of the form of (1)).

The kinetic model is fit to experimental data about the system by assigning values to a set of parameters in the equations of the model. Among these, the most widely used are reaction constants and affinities.

Once the parameters of the system have been established, we can describe important qualitative and quantative biochemical characteristics of the system such as stability, reaction time constants, etc. by examining the behavior of the model. Furthermore, we can use the model to attempt to predict in terms of it, reaction-time courses for different initial conditions than the ones under which the system parameters were found and with the aid of some (mostly empirical) additional relationships we can predict the effect of variations in environmental conditions (e.g. temperature) on the dynamics of the system.

The techniques of macrokinetic theory described above, have been adapted for the study of functional operation of cellular and subcellular systems by many researchers in the field. A sample of this approach can be found in the works of Banks [8], Davidson [9], [10], Goodwin [11] and Rosen [12], from the theoretical point of view, and, Hood, Wilson and Wood [13] and Goodwin and Ziegler [14], from the experimental point of view.

Although macrokinetics has been found to be a useful tool in the study of functional aspects of the genetic apparatus of procaryotes, (e.g. Reiner [26]), it has important shortcomings in these tasks that are worth discussing here, since they provide motivation for studying these systems with a different modeling strategy.

The first limitation, and in our opinion, the crucial one, was already mentioned previously; namely, the fact that given the reactions characteristic of the system (a piece of data that is not always available) we have to assume the reaction structure of the system in order to obtain a model of the type described earlier. This, of course, means that although the model derived in this fashion and the system coincide in their behavior for a given set of initial-component molar concentrations, their dynamical structures[1] may be widely different and consequently functional, structurally-dependent properties of the elements obtained on the basis of an analysis of the model are, at the very best unreliable and frequently totally erroneous (Bartholomay [15]).

The reason behind this limitation, as pointed out by many authors (e.g. Higgins [16], Rosen [12]) lies in the fact that given the reactions characteristic of a system of molecules such as the genetic apparatus of a procaryote, the structure of the dynamics of the system, has from the macrokinetic point of view, a great number (infinite, in fact) of feasible reaction scheme structure models representing it.

We mention that biochemists ([15], [17]) are aware of this limitation and have developed a new theory, called microkinetic theory, which is based on quantum mechanics and whose main objectives are to study the structure of the dynamics of systems of molecules by determining the dynamic structure of each of these molecules in terms of the properties of the atoms that form them and the interaction-characteristics between molecules as a consequence of these properties.

---

[1] That is, the dynamical structure of the model and that of the system it attempts to represent.

In principle, given a system composed of N molecules such as the one sketched earlier, if we can determine its microkinetic model (which boils down to determining the Hamiltonian[1] operator of the system) we can determine (up to the uncertainty level of the system) all the structural characteristics of its dynamics and therefore, the functional roles played by each type of molecule in the system. However, microkinetic models, even for the simplest system (e.g. a monomolecular-single reaction system), are so complex from the mathematical and computational points of view that even with considerable approximations such as slater energy hypersurface approximations (Slater [18]), the analysis of the functional characteristics of the model is very complex if not impossible.

Nevertheless, microkinetics theory is a very promising area of research and its application to the detailed study of genetic systems will give, in the future, a better comprehension of the characteristics of these systems. We point out that microkinetics theory has been used successfully in the study of some morphogenetic properties of macromolecules[2].

A second limitation of macrokinetics in the study of genetic systems is related to the methodologies for gathering experimental data. These techniques, with very few exceptions, require in-vitro experi-

---

[1] See Bartholomay [15].

[2] See Turing [53].

ments ([15]), in which only a small subset of the reactions of the system are identified. The implicit assumption here is that the effects of the rest of the system on the subset of reactions under experimentation, is negligible or at least can be phenomenologically accounted for. It has, however, been experimentally proven in a number of cases (see Lewin [19]) that in-vivo time-course behavior of subsystems of genetic systems is very different from the in-vitro behavior of these subsystems.

Finally, a third limitation, also related to the experimental techniques of macrokinetics, involves the relation between the time and spatial scales of these experimental techniques and the time-horizon and spatial extent of the genetic system under study. These techniques usually involve measurements that are carried out, not on an individual cell or subsystem of a cell, but in aggregates of cells (cultures) or subsystems, and the time-interval of measurement spans several generations ([20], [21]). These experimental conditions are required in order to ensure statistical confidence of the measurements. However, the objective of the experiments is to identify parameters of reactions whose time constants are shorter by several orders of magnitude than the time interval of a single generation. This implies that the resultant parameters correspond to a smoothed average (over time and space of the time-course of the system. Consequently, important structural variations of the system are not reflected in the resultant model. (An example of this effect is analyzed by Holland [22], with respect to

epistatic effects in catabolic repression.)

In the last paragraphs we have made a brief critical description of some important aspects of macrokinetic and microkinetic techniques with emphasis on their applicability and limitations to the study of genetic systems. Our purpose in this exercise was to establish a frame of reference for comparison purposes with our study of epigenetic control mechanisms in procaryotes which we will formulate next.

The first task towards the formulation of our problem consists of defining what we mean by epigenetic control mechanisms. This is necessary due to the fact that this terminology is used for designating different (although related) concepts in cellular genetics, population genetics and cellular physiology (see for instance Waddington [23], Goodwin [24], Kauffman [25] and Reiner [26]).

By epigenetic control mechanisms in the genetic apparatus we mean a finite set of temporal and/or spatial interactions among the chemical species (and organelles) present in the cellular space, by which the activity of the cell is regulated and modulated, based on the information stored in the genes of the cell.

Admittedly, the definition of the last paragraph is ambiguous and perhaps too general; our purpose in the following paragraphs is to qualify its major components by formulating, in general terms, the methodology we use in this study, and discussing in terms of it, the objectives of our research.

As in most studies in modern biology, ours is based on the develop-

ment of a model of the genetic system of the procaryotic cell. The fundamental characteristics of our model are the following:

1. The model is an ensemble of elements each one representing a particular class of molecules in the system. The number of different classes of elements in the model is finite.
2. The elements of the model in each class represent the function of the respective class of molecules in the genetic apparatus. That is, the time-course evolution of an element of the system represents the activity of a molecule (or organelle) in the system that fulfills a specific function in the cell (e.g. genetic storage, catalysis synthesis of control-elements, etc.).

The second set of characteristics motivates our choice of models for the different classes of molecules in the apparatus:

3. A typical molecule in the system is represented by an element whose dynamic structure is capable of processing finite amounts of information (whose characteristics will be described later). This information is provided dynamically to the element by interaction with other elements of the model and by its internal storage (i.e., an element has memory).

In this representation of molecules of the genetic apparatus, we do not model their physico-chemical characteristics, but their functionality. (Of course, this functionality is determined by their physico-chemical properties; but in the model these are not explicitly considered.)

This modeling approach is based on one of the fundamental conclusions achieved in a symposia in Villa Serbelloni (Italy) in the Fall of 1965 under the chairmanship of Waddington (see, Waddington [26a]). (The objective of the symposia was to establish the basis for a coherent development of theoretical biology as a scientific discipline.) The alluded to conclusion was that biological system dynamics, in general, and cells in particular, have, from the functional point of view, great similarity with the evolution of computational processes in digital computers. (In particular, the papers of Arbib [27], Fraser [28], Lewontin [29], Burns [30], Wolpert [31] and others presented at the symposia explore several variants of this analogy.)

Although we don't use this analogy as a working principle throughout the development of our model, it certainly constitutes a frame of reference to which we go from time to time to assess our results in a computational context. For instance, in 3.) above we considered, at the element level, the dynamic structure of an element as determined by its observed (or assumed) operational characteristics and not its physical properties (which determine those characteristics); this approach is commonly taken in the study of computer systems in which a computational process can be described by specifying a set of instructions (the program) and the time evolution of the variables of the process (as determined by the program) without any knowledge of how these instructions are physically implemented (e.g. see Engeler [32]).

4. In addition to representing molecules (i.e., representing their function), we must provide our model with a formal description of the interactions among elements. Since the elements represent classes of molecules, this amounts to specify the mechanisms of information transfer between typical elements of the different classes and among two or more elements of the class.

From the operational point of view, these mechanisms fulfill two tasks: (a) they provide the means by which information is transferred among elements of the model; and (b) they determine how an element selects dynamically the elements with which it interacts.

Task(b) above requires for each element in the model to possess internal computation capability to perform the selection of the elements with which it interacts.

5. In 3.) and 4.) we established some general properties of the elements of the model without specifying where and how these elements are spatially allocated. In studying the genetic apparatus from an operational point of view, this issue is of fundamental importance because the time-course of genetic processes is determined by flow of information among the elements participating in these processes (see Levin [34] for an in-depth analysis of this aspect) and this flow is accomplished by physical migration of molecules in the cellular space (e.g. mass-diffusion in the intracellular and extra-

cellular spaces), and/or by wave-like propagation or dispersion of disturbances (e.g. local pH variations in the intracellular space, electric pulses in the cellular membrane, etc). Thus, an adequate representation of the cellular space, from the operational point of view, requires not only a coordinatized set to represent the physical space, but an effective way for representing the information flow throughout the system.

In other studies of operational aspects of epigenetic control mechanisms ([24]), the information-flow is represented by defining signals which carry the information between the elements of the system. In our study, we do not take this approach. Based on an analysis of the operational model devised by Jacob and Monod [35], which have been corroborated by many experiments ([36]), we represent information flow in the model by well defined changes in time of the internal storage of the elements (as we shall see; changes in the <u>state</u> of the elements). In each element, these changes are functionally dependent, at each time, in the information stored in the elements which are located in the <u>nearest-neighbor coordinates</u> of this element, in a space whose coordinates are defined in such a manner, that at any time, an element allocated at a given point of the space, <u>interacts with, at the most, one element along each coordinate</u>.

We note that under the representation of the space in which the model operates, suggested above, there is no actual motion of the elements (for information transmission purposes) but rather, the informa-

mation flow is simulated by dynamic changes of the state of the elements of the model which are functionally dependent on the states of corresponding neighboring elements.

We refer to the representation of the space of the model outlined above, as the Informational space of the model.

6. The previous characteristics of the model refer to dynamic-operational properties of the model; now, we state some properties of the model which refer to the control aspects of the genetic apparatus.

Perhaps the most evident characteristic of the genetic apparatus from the control point of view, is the absence of a central controller (although there is a central storage of genetic information). The control activity of the genetic apparatus is carried out usually by several processes more or less independent of each other, whose regulation is accomplished by the local interaction (in time) of the elements participating in the process.

A second characteristic of the genetic apparatus with respect to its control activity, refers to the coordination[1] of processes; this task is also accomplished locally (i.e., at the element level); that is, there is no central coordinator for the processes of the system. In order to satisfy the requirement in our model, we must provide the ele-

[1]By process coordination we understand a set of mechanisms by which two or more processes time their interaction (see Chapter 4).

ments with appropriate mechanisms so that the coordination of activities is implicit in the resultant operational rules of the elements. These rules are described later on in this section.

Finally, all the control actions for the operation of the system are generated within the system and carried out by interaction among its components. That is, there is no external controller.

7. One of the most important operational aspects of the genetic apparatus, which has received a great deal of attention over the last 10 years ([37], [38]), concerns the interrelation between different processes which are being carried out simultaneously by the system. It turns out that these processes are well ordered with respect to at least two criteria; time and structure (also called spatial organization). The resultant classifications of processes according to these two criteria are called time and structural hierarchies, respectively. (see Pattee [37])

Process hierarchies play a very important role in the analysis of control strategies in the genetic apparatus, as we shall show in terms of our operational model in Sections 2.3 and 4.5. Here, we mention two aspects of the dynamics of the system, which are critically dependent on the hierarchization of the processes involved in these dynamics. These are, resource allocation and coordinated response to environmental disturbances.

By resource allocation we mean the dynamic distribution (sometimes

called sequencing (Ziegler [39])) of elements of the system which participate in more than one process and whose numbers might be in short supply. (For instance, ribosomes are used in every protein synthesis process, also, in some reactions, two or more elements compete for the same substrate.) We will see that this allocation of elements is implicit in the characteristics of the processes in the system and depends on a hierarchical ordering of processes that we call time-hierarchy, which is established as its name indicates, by time-dependent parameters which are associated with each process in the system.

In the coordinated response to environmental disturbances by the genetic apparatus, it is common to have several processes, evolving simultaneously, which do not interact among themselves during most of their time-courses except at a specific instant of time, at which, they interact through some of their elements. The mechanisms of this interaction will be analyzed in terms of the model, in Chapter 5, where we will show that this coordination is a function of the time hierarchy introduced in the last paragraph. (An example of this coordination can be observed in the processes that form the metabolic cycle in the cell, see Watson [33].)

In addition to a time-hierarchization of the processes that constitute the response to a disturbance to the system, another ordering of the processes that participate in its dynamics is observed. This ordering classifies the processes on the basis of their control action, thus, it is called control hierarchy.

In the control hierarchy a given process P is said to be of lower hierarchy than a process P^1, if P in order to fulfill its function, requires information from process P^1. The information from P^1 to P is transferred via interaction of some of the elements participating in these processes.

We note that as characteristics of the genetic apparatus, both time and control hierarchies have close analogs in digital computer systems. The time hierarchy introduced above, corresponds to the on-line priority assignments in parallel synchronous processes (see Stone [40]), while the control hierarchy corresponds to the structured organization in multilevel-process computers. (see Tannenbaum [41]). We take advantage of this similarity when we translate the properties of the system related to these hierarchies into our model.

8. One of the most remarkable properties of the genetic apparatus is the fact that the structure of its dynamics changes as the system evolves in time and space, as a function of its state.[1] These changes are not just parameter changes, but they involve the controlled construction of subsystems that perform a preassigned task. In this study we refer to processes whose control activity consists of the construction of subsystems on which processes are executed, as construction

[1]The process of catabolite repression provides an example of this characteristic (see Hayes [54]).

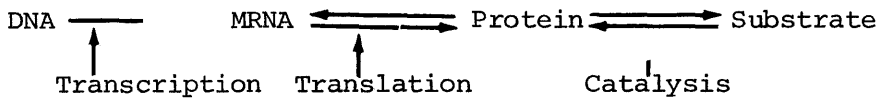
processes. An obvious example of construction processes is given by the controlled (as opposed to constitutive) synthesis of proteins (see Lewin [36]).

The previous 8 attributes of the genetic apparatus and of our modeling philosophy are the basis on which we have developed the operational model of epigenetic control mechanisms, the main subject of our study.

The remaining part of this section is devoted to a description of this model and a comparative analysis of its characteristics with those of the macrokinetic model sketched before.

The microkinetic model, or rather, a particular representation of it, will be used in Chapter 3 as the basis under which our operational model is constructed.

The eight attributes are characteristic of the following operational principle known as the (reduced)[1] dogma of cellular genetrics (see Crick [42]).



The dogma establishes the flow of genetic information in the system and thus determines the operational characteristics of the control mechanisms in the genetic apparatus.

[1]The dogma presented here is a reduced version of the one commonly accepted today in which DNA is modifiable under inverse transcription from M-RNA.

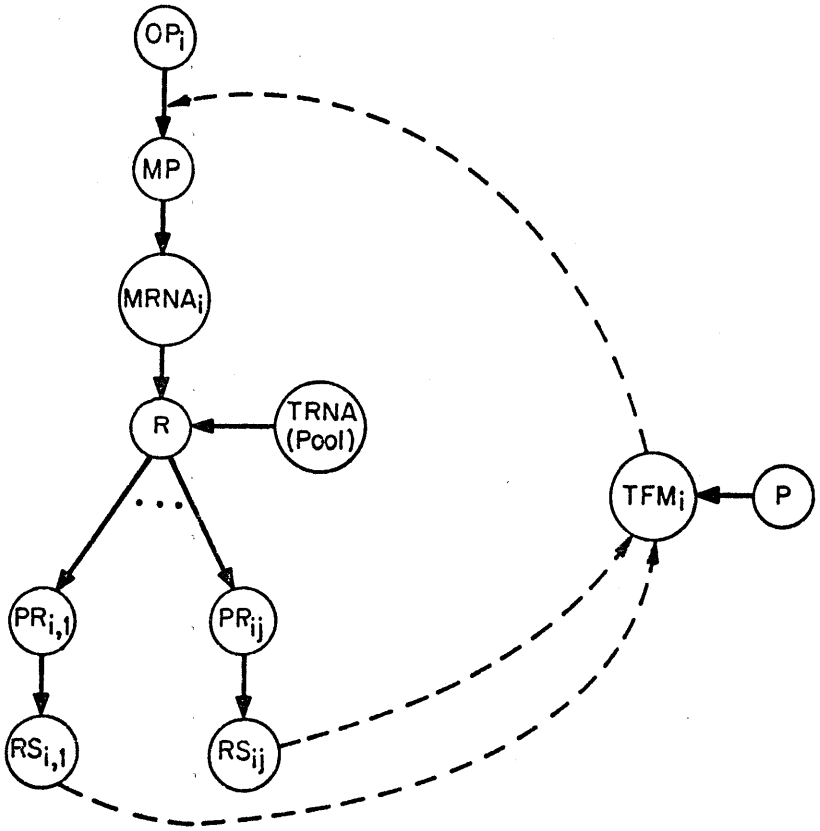
Briefly, the incidence-diagram above indicates that the control actions of the genetic apparatus are executed by catalytic proteins which are synthesized by a translation procedure under the direction of a MRNA encoding of the amino acid residues forming these proteins, and this MRNA molecule is transcribed from a specific locus of the DNA of the cell.

The information flow suggested by the diagram above is not a spontaneous process, but it is triggered by specific conditions in the system (e.g. the presence or absence of a given metabolite(s)). Figure 1 gives the incidence-diagram of a simple epigenetic process in which the information flow proceeds according to the dogma.

In Figure 1 OP_i stands for an element representing the portion of DNA of the cell coding for the ith operon[1] (see Figure 2). MP represents the element that carries on transcription (a protein called MRNA polymerasa). $MRNA_i$ is the transcribed version of the operon, R is the element that translates $MRNA_i$ into one or more proteins, represented in the figure by $PR_{i,k}$ $k=1,\ldots,j$. These proteins catalyze (or inhibit) specific reactions which are denoted by $RS_{i,k}$ $k=1,\ldots,j$ and whose representations in terms of elements will be discussed in a moment.

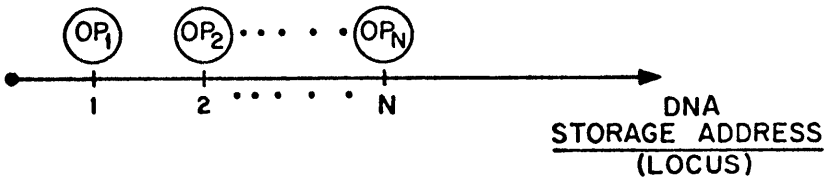
Finally, TFM_i represents the transcription control mechanism for the ith operon. The TFM_i has as its function to establish the operational interaction between the control metabolite element, the regu-

[1]In this study an operon is a set of genes under a common transcriptional control. E_x lac operon. (see Lewin [36]).



CONVENTIONS	
OP_i	- i^{th} operon of the system
MP	- MRNA polymerase
$MRNA_i$	- MRNA transcription of i^{th} operon
R	- Ribosome
$PR_{i,k}$, k= 1, ...j	Proteins of the i^{th} operon
RS_{ik}	- Reaction system controlled (catalyzed) or inhibited by the k^{th} protein of the i^{th} operon
TFM_i	- Transcription feedback mechanism for the operon
P	- Regulatory protein

Figure 1



REPRESENTATION OF DNA OF THE CELL AS A SEQUENCE OF ELEMENTS REPRESENTING OPERONS. EACH ELEMENT IS LOCATED IN A SPECIFIC LOCATION (ADDRESS) AND THESE ARE NOT EXCHANGEABLE

Figure 2

lator protein and the operon, as will be explained later. We note that in contrast with the other elements in the epigenetic control mechanism of the operon, TFM_i does not represent a molecule, but a control action; for that reason these elements, as well as other elements of analogous characteristics that will be introduced throughout our study, are referred to as virtual elements.

Before passing to the description of the representations of the reactions in terms of elements we digress for a moment to discuss the meaning of the subindices of the different elements in Figure 1.

First, the subindex i in OP_i indicates that this element represents the operon located in the ith address of DNA. It is well known (see Watson [43]) that every gene (and consequently every operon) occupies a specific location in the DNA of procaryotes, therefore the map

$$\text{Addresses} \longmapsto \text{Operons} \tag{2}$$

is 1-1 (in biology literature the set of addresses is called the locus of the genes, see Hood, Wilson, Wood [13]).

Further, we shall assume that the number of different operons (N in Figure 2) in a given cell under study is an invariant of the system; therefore (2) is also onto.

The second subindex in $PR_{i,k}$ indicates that the corresponding element is representing the kth protein coded by operon i. Since proteins of a given operon are synthesized sequentially, this index carries also time information as indicated in the diagram of events shown in

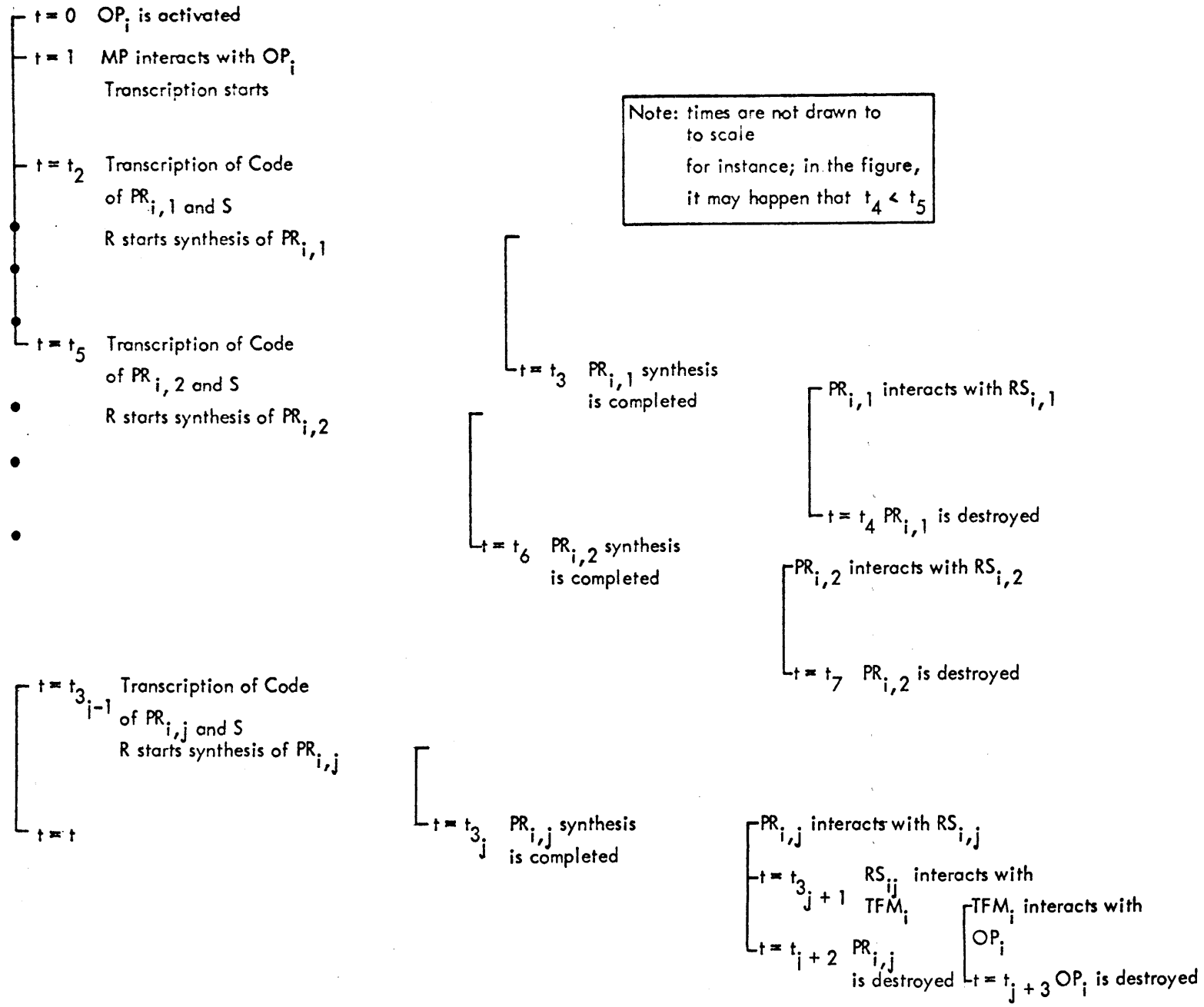


Figure 3 Diagram of Events for the System of Figure 1

Figure 3.

Figure 3 displays the important events in the evolution of the epigenetic system of Figure 1. We note that we have idealized two important aspects in the evolution. First, we have assumed that every protein of the operon is not synthesized until its MRNA is completely transcribed and second, any protein does not start its catalytic action until completely synthesized. There is a considerable amount of experimental evidence (e.g. Malkin and Rich [44], Lewin [36]) that in both cases the action of these elements is initiared while their transcription or synthesis are not completed. However, for the purposes of our study these characteristics do not represent crucial operational properties and therefore these assumptions do not affect, significantly, the results of this study.

Notice also that in Figure 3 we have indicated that $MRNA_i$ and $PR_{i,k}$ have finite life spans, a feature that has been experimentally verified[1] and is clearly desirable for the epigenetic mechansim to respond to actual conditions of the cellular environment it is regulating.

Figure 4 illustrates the characteristics of a subset of the most important reaction system structures that we consider in our model of epigenetic mechanisms. We note that in each of the cases the reaction

[1]See Lewin [36].

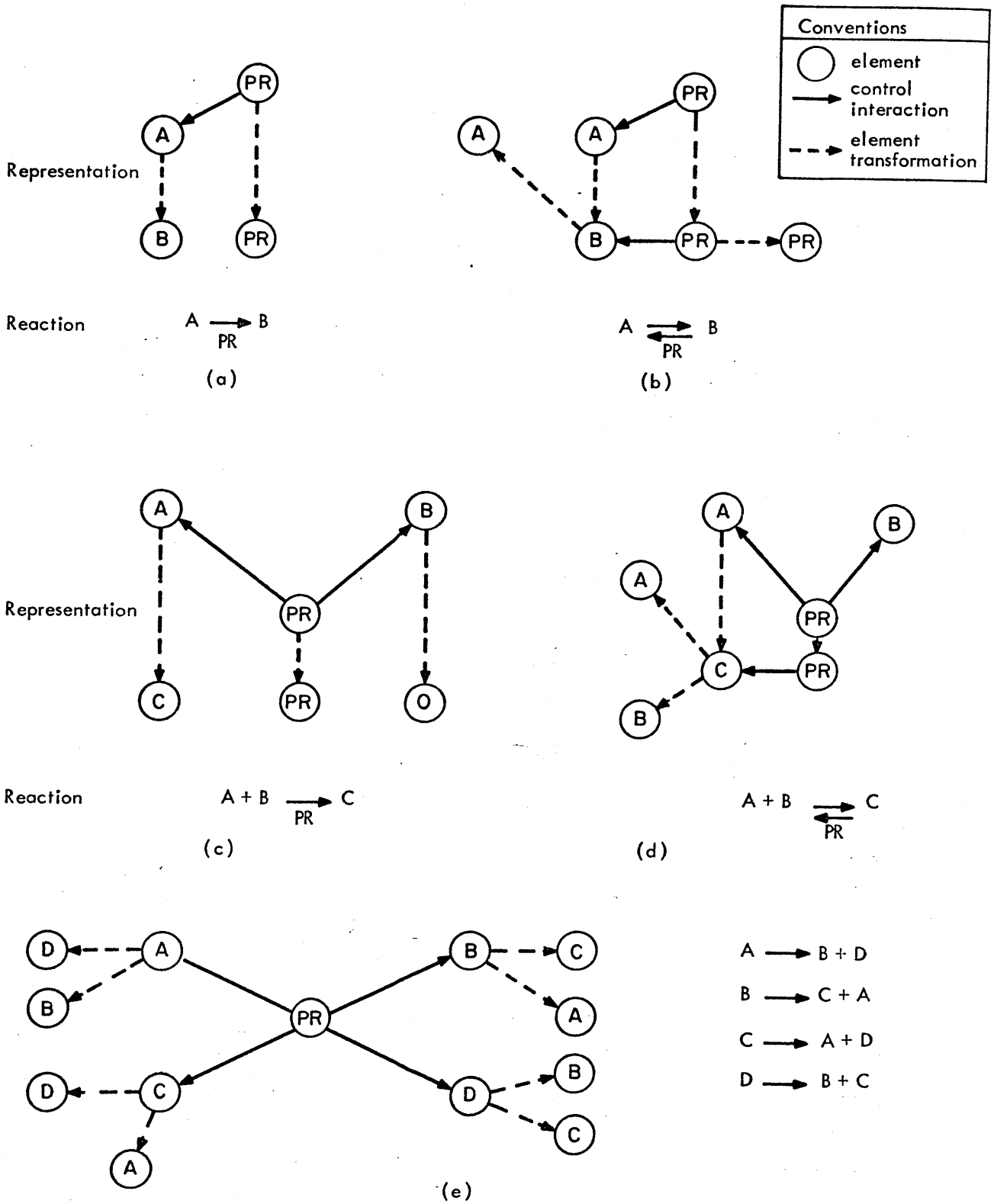


Figure 4 Important Reaction System Representations

is assumed to be catalyzed by a protein. Although every reaction that evolves under catalysis can evolve autonomously (with a much smaller reaction constant), it is safe to assume (see Watson [33]) that in procaryotic cells all the reactions evolve under catalysis.

The simplest reaction system is the one corresponding to the monomolecular irreversible reaction illustrated in Figure 4a. In our model, we represent the evolution of this reaction by assuming initially that two elements; A representing the reactant and PR representing the catalytic protein, interact in a manner that will be explained later. As a result of this interaction, element A becomes element B, the product of the reaction.

The transformation of an element representing A into element B, mentioned above, involves, as we shall see, a state transition of the element. This transition is driven by the interaction with the element representing PR. We note also from the figure that the element representing PR preserves its identity, a feature that is in accordance with physical evidence. (see Bartholomay [45]).

In Figure 4b there is an illustration of the representation of a monomolecular reversible reaction. No attempt is made in the figure to represent timing of the events $A \xrightarrow[PR]{} B$ or $B \xrightarrow[PR]{} A$ only their occurrence. For the timing of this as well as other events of reversible nature in the apparatus we will include in our model a class of elements which, as with the transcription control mechanisms discussed previously, do not represent molecules, but rather actions. In this case they represent

time coordinating actions. We refer to these virtual elements as clocks.

Figures 4c) and 4d) illustrate the basic aspects of biomolecular irreversible and reversible reactions respectively. The representations for these reactions follow similar patterns to their respective counter parts for monomolecular reactions. They are explicitly included in the model because as shown in Section 3.3, the behavior of the latter differ radically from the behavior of the former, from the operational point of view.

In Figure 4c) the symbol "O" indicates that the element has reached a special state called dormant state whose meaning is no information is carried by this element. We clearly need such an element to describe what happens to elements in reactions with fewer products than reactants. We think of some of the reactant elements as being "transformed" to the product elements, while the other reactant elements "vanish" -- i.e. they reach the dormant state. The dormant state and its meaning will be discussed further in Chapter 3.

The representation of multimolecular reactions of the form,

$$m_1 A_1 + \ldots + m_k A_k \underset{PR}{\overset{\rightarrow}{\leftarrow}} n_1 B_1 + \ldots + n_\ell B_\ell \qquad (3)$$

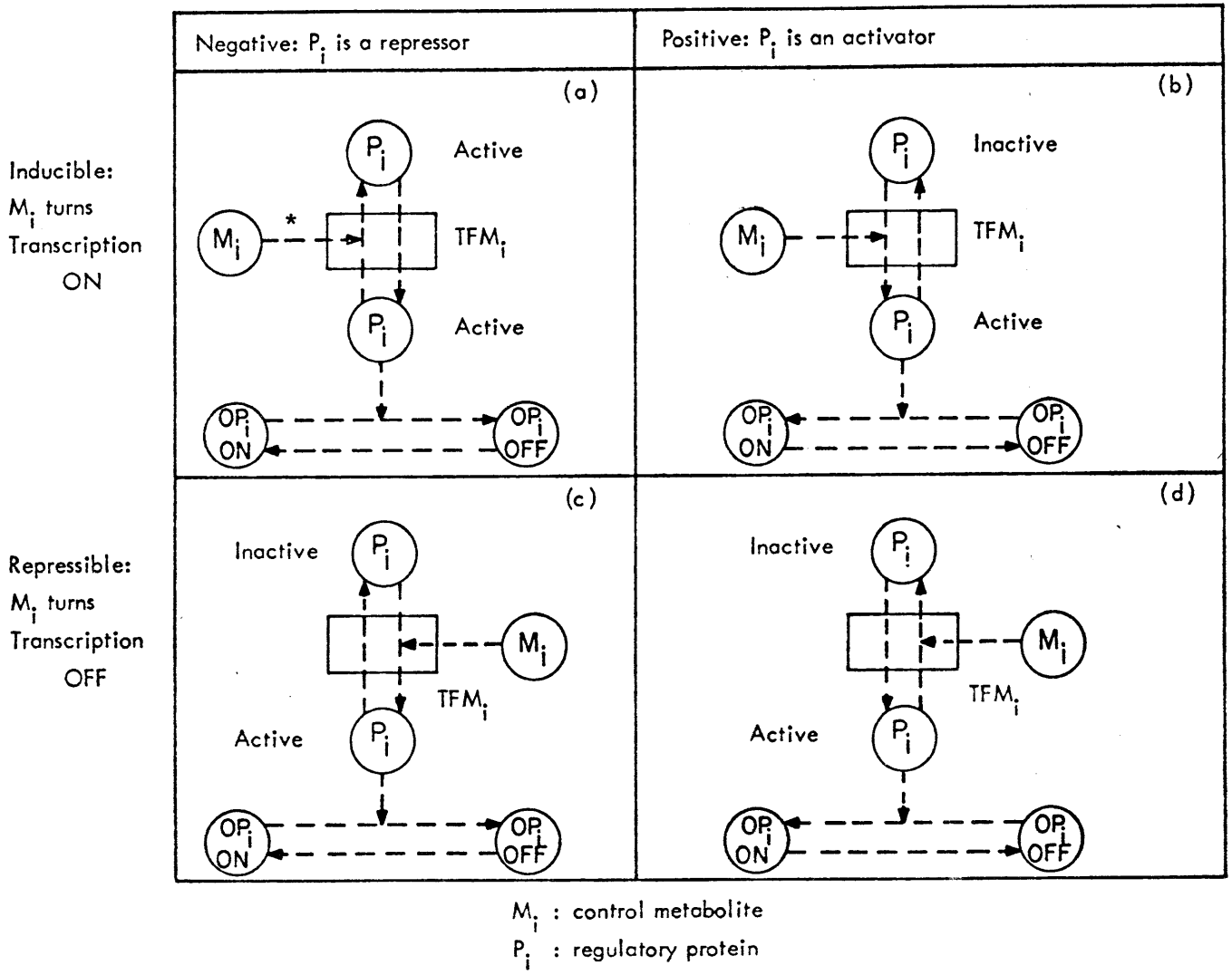
where m_i, $k_i < k$, n_i $1 \leq i \leq \ell$, are stoichiometric coefficients, will be carried out by expressing (3) as a sequence of reactions, each one of them of one of the forms indicated in Figures 4a) - 4d). Aside from the fact that this provides us with a manageable representation of

very general reaction systems (i.e., reaction systems involving several coupled reactions of the type (3)), it is frequently the case that reactions of the form of (3) do evolve in steps of the type of Figures 4a)-4d).

Finally, Figure 4e) illustrates an important type of reaction (sometimes called a ring or cycle) which, as we shall show, exhibits remarkable oscillatory characteristics (the various elements appear and disappear at constant time intervals). Reactions of this type will be proposed in our study as the processes that provided the cell, and in particular the genetic apparatus, with a time base for its control actions. (Reactions of these and similar structures are sometimes called biochemical clocks (Prigogine [46]).)

Now, to complete our brief overview of the epigenetic control mechanism depicted in Figure 1, we discuss some important characteristics of the virtual element representing the transcription control mechanism. It is well known (see Jacob and Monod [35]) that the operation of the mechanism depends on the characteristics of interaction of two elements: the controlling metabolite (M) and the regulator protein (P). Each one of them can operate according to one of two schemes (see Figure 5). This implies that we must consider four possible types of different transcription control mechanisms (TFM_i); as illustrated in Figure 5.

According to the controlling metabolite (M), which in our example of Figure 1 is one of the reactants or products of one of the reaction systems, (this provides the feedback), (TFM_i) can be of one of two



M_i : control metabolite

P_i : regulatory protein

*The arrows here indicate transitions that occur. For example, in (a) the presence of M_i transforms P_i form active to inactive

Figure 5 The Classes of Transcriptional Control Mechanisms

types; Inducible is M turns the transcription on,and Repressible if it turns the transcription off. Notice that M is an input to TFM_i (i.e., as we shall see its state is an input).

According to the regulatory protein (P), which need not be (and usually is not) a protein of the operon (OP_i) it is regulating, TFM_i can be one of the two types; Negative is P represses transcription when present and Positive if P activates transcription when present.

Combining these two characterizations we obtain the four classes of different transcriptional control elements as illustrated in Figure 5. Examples of operons corresponding to Figure 5a) in E coli include lac and gal operons, to Figure 5b) ara and mal, to Figure 5c) trip and his (see Watson [33]). We could not find an example for an operon controlled by a transcription mechanism corresponding to the representation of Figure 5d); however, we find no reason from the operational point of view for such an operon not to exist and we shall illustrate with a hypothetical example, a case in which this type is the best type of control (see Section 3.5.1).

In general, epigenetic control processes in procaryotes involve several coupled operon systems as indicated in Figure 6, in which we display a 3-operon system involving operons OP_i, OP_j and OP_k. The regulator protein of OP_i is coded by OP_j and its control metabolite belongs to reaction system $RS_{k,l}$.

For operon OP_j transcription is regulated by protein PR_{k2} which also regulates the transcription of OP_k. The metabolite controlling OP_j

belongs to $RS_{i,1}$.

Finally, operon OP_k is regulated by a protein synthesized by itself (as we shall see in Section 3.5.1 this implies that TFM_k must be of the repressor type), $PR_{k,2}$ and controlled by a metabolite belonging to $RS_{i,1}$.

The last example illustrates the complexity of the structure of epigenetic control mechanisms. We note that this complexity, measured as a function of the number of elements evolved in the process, or the number of links in the incidence diagram of the process is considerable even for the case in which only a few operons are involved. The example of Figure 6 also illustrates important operational properties that in certain measure are generic to all epigenetic control processes. These properties, known as self instructive catalytic properties were suggested by Eigen [47] as the means by which global coordination of multiple operon systems is achieved. In section 3.5.1 we will implement the example of Figure 6 in terms of our model and in Chapter 4 characterize algebraically its self instructive catalytic properties.

We note that, to our knowledge, the system of Figure 6 does not correspond to any known "real life" epigenetic mechanism, but it possesses many of the characteristics observed in some of the operons systems in E coli (Lewin [36]) and is simple enough so that these characteristics can be studied with a reasonable sized model.

In the last paragraphs we have given a diagramatic description of some characteristics of the epigenetic control mechanisms from the operational point of view, with emphasis on the transcriptional con-

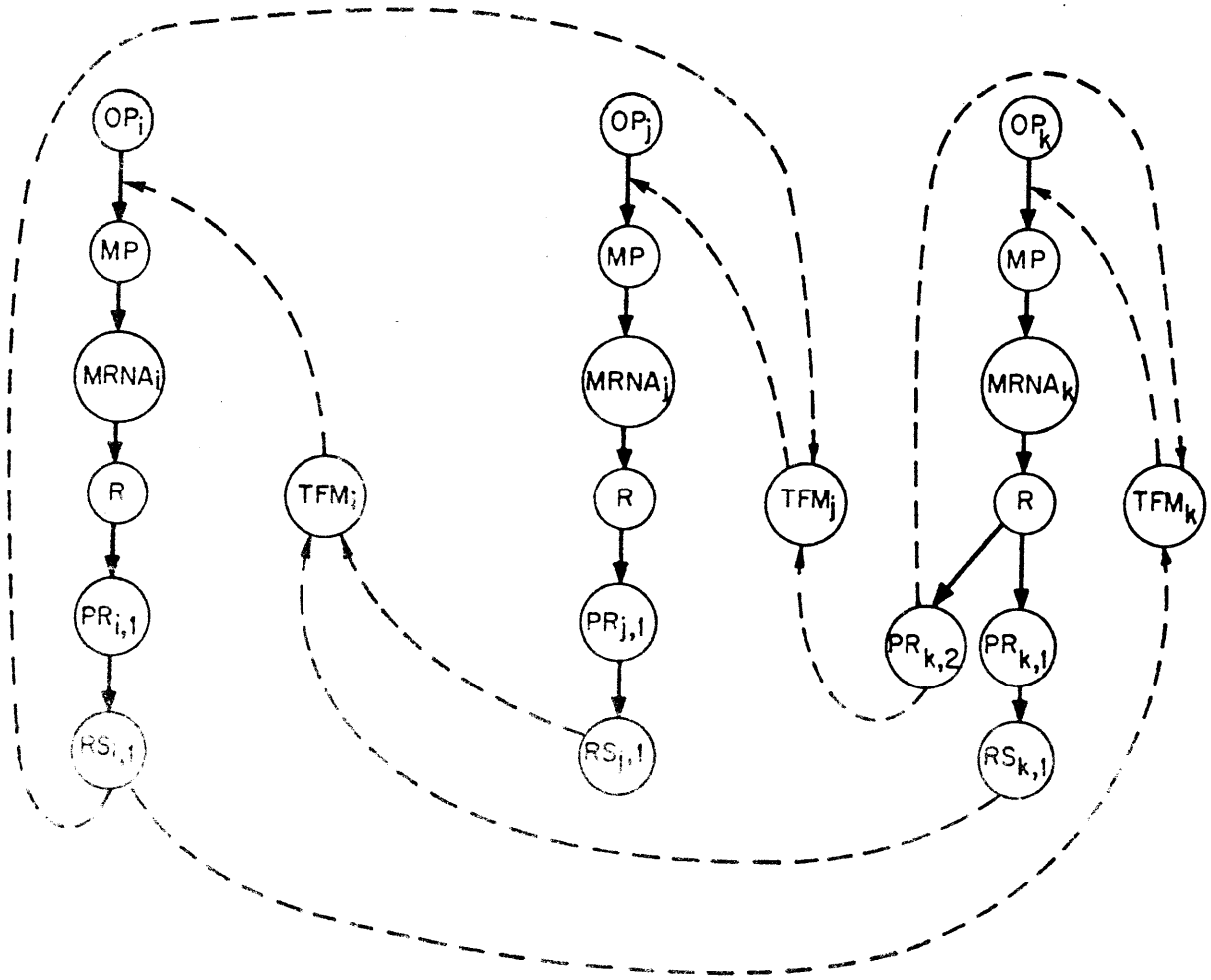


Figure 6 Three-Operon Epigenetic Control Mechanism

trol. The overall structure of these controls, as suggested by Figures 1 and 6 is based on the model for the lac operon in E coli originally proposed by Jacob and Monod [48], which has been shown to agree with experimental data for other operon systems [36].

We note that the version of the model of Jacob and Monod given above, on which our study is based, is an ensemble of specialized interacting elements, some of them representing actual molecules that appear in the cell and some representing activities rather than physical elements.

The dynamics of the system is represented by transformations in the state of its elements and these transformations are driven by local interactions among elements. This characteristic is in contradistinction to the macrokinetic model discussed earlier in which the dynamics of the system are represented by assigning an intensive variable (concentration)[1] to each molecular species in the system and then, establishing,more or less phenomenologically, a set of mass-balance (and/or energy balance) expressions for reaction-schemes among these molecular species. These expressions are then used to write down the dynamic equations representing the evolution of the system, with the implicit assumption that some of its important operational characteristics will be captured by the model derived in this fashion.

---

[1]An intensive variable is defined at every point in the cellular space. Thus for instance, we talk about the concentration of a protein and assign to is a value at each point in the cellular space despict of the fact that only a finite (integer) number of proteins can exist in the space at any time. In our model, an element represents a molecule in a specific location on the space, therefore we do not have intensive variables describing the dynamics of the interaction between elements.

In synthesis, in this section we have established a set of general principles on which our modeling strategy is based. We have illustrated diagramatically the interaction-structure among the elements that form this model and have described some of the important operational characteristics of these elements in relation to the molecules or epigenetic actions they represent.

2.2 Description of the Concept of Distributed Dynamics and Neighboring Interaction

This short section is devoted to the introduction of two fundamental dynamic properties of the genetic apparatus in procaryotes, that have determined in great part the structure of our model.

In ths last section we established that our model consists of an ensemble of elements and the dynamics of the system is determined by interaction (in time) of these elements in a coordinated space that we called informational space. In this section, we will define this space and describe how the elements of our model are allocated to it.

We assume throughout this study that the elements of the model have null volumetric extension in the informational space. Since our purpose is to study operational characteristics of epigenetic control processes which are dependent on the functionality of the elements participating in these processes, and not on the physical structure of these elements, the assumption is not a limiting one.

Based on the previous assumption, we now explain how elements are allocated in the informational space. The idea is to assign to each

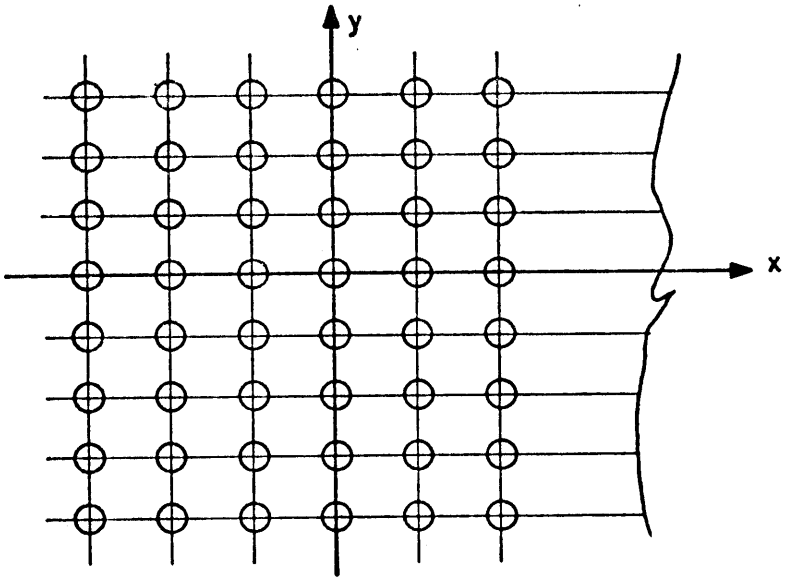
ordinate point of the informational space one and only one element. For most of this study the informational space will be the cartesian product of k copies of the integers (Z^k) where $k \geq 1$, is an integer chosen according to the criterion described in the next paragraphs.

The discussion above gives us an additional characteristic of the model which can now be defined as an ensemble of interacting dynamic elements (whose characteristics will be introduced in Chapters 3 and 4), allocated in a k-dimensional grid. This is illustrated for k=2 in Figure 1.

The dimension k of the informational space is chosen according to the following criterion:

We consider epigenetic control processes of the type diagramatically illustrated in Section 2.1 (Figures 2.1.1 and 2.1.6), and note that any given element interacts directly with a finite number of elements at any time. (Interactions are indicated by the arrows joining the elements.) For the purposes of analysis and to stick as close as feasible to the characteristic of the genetic apparatus, it is convenient (but by no means necessary) that at any time t, the elements interacting with a given element M to be allocated at this time in the nearest-neighbor coordinates of the coordinates of M in the informational space. Our model has been developed according to this constraint. That is, no interaction at-a-distance between elements is allowed.

We note that this requirement for the model, plus the fact that at any point in the informational space we have a single element, imply that the maximum number of interacting elements with a given element,



Two-dimensional (k=2)

Informational space circles represent elements

The informational space is unbounded and has one element at each point of its points

Figure 1

at a given time t, in a k-dimensional informational space is 2k+1 (that is, including the element itself). Therefore, we choose k according to:

— criterion for choosing the dimension of the information space (k)

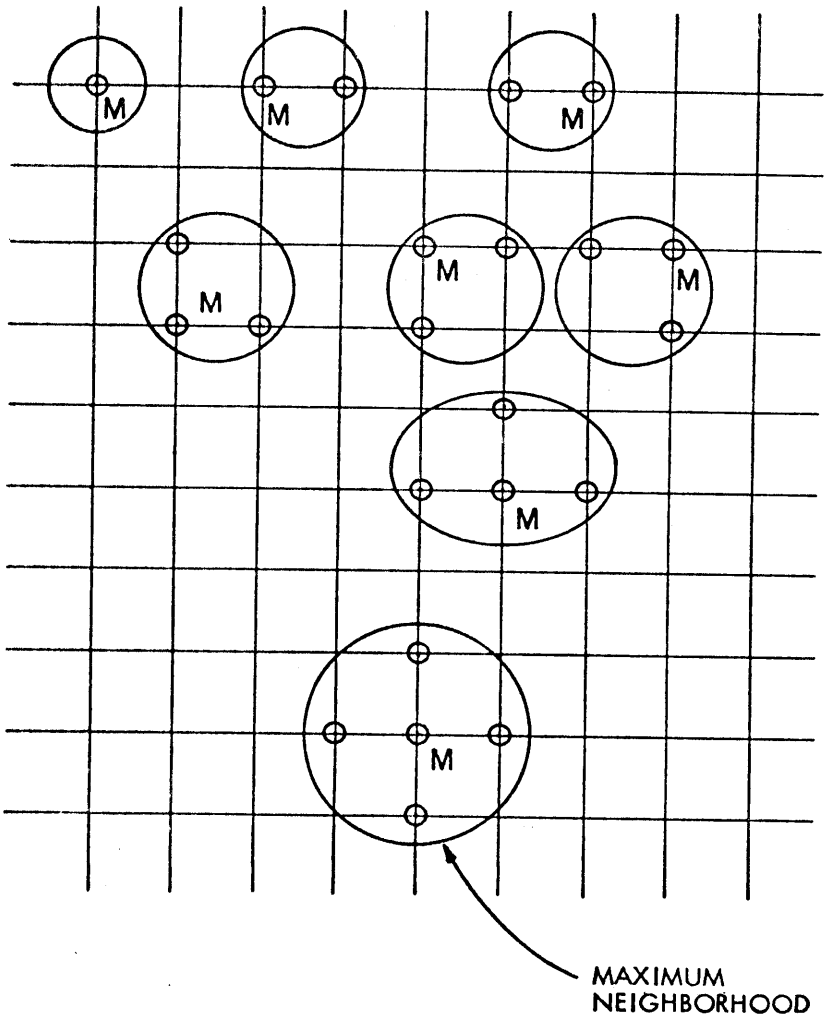
2k+1 = maximum number of interacting elements at any time in the epigenetic control mechanisms of the system under study. Δ

Clearly, not every element in the model will interact with the maximum number of its nearest neighboring elements (2k+1). Figure 2 shows a subset of the possibilities of nearest-neighbor interaction in a 2-dimensional informational space.

It is easy to see that in a k-dimensional informational space, there are 2^{2k+1} possible nearest-neighbor arrangements. We will see in Chapter 3 that only a small subset of these have meaning in our model.

We now establish some important aspects of the elements of our model which are consequences of the nearest-neighbor interaction requirement.

As mentioned above, the dimension of the informational space is chosen so that the maximum number of elements invloved in an element interaction (2k+1) are nearest-neighbors in the space. This implies that in any interaction which involves $m < 2k + 1$ elements, the corresponding element must possess internal logic capable of choosing the appropriate m elements with which it interacts. We will see in Chapter 4 that this task is accomplished by a subsystem that forms part of



A Subset of the Possible Nearest-Neighbor Arrangements
in a 2-Dimensional Information Space

Figure 2

every element of the model that we referred to as Input selector.

The input selector of an element is a device that selects as a function of the state, one of the possible 2^{2k+1} nearest neighboring arrangements that we refer to herein as neighborhood functions and whose characteristics are described in Chapter 4.

In a given element, once the appropriate neighborhood function is selected, the states of the corresponding nearest-neighbor elements constitute the input that drives the element, as discussed in the last section.

Finally, we note that the informational space we have chosen for our model is unbounded and since at each of its points we have assigned an element, the number of elements in our model is infinite. This characteristic does not satisfy one of the constraints we established in the last section, namely that the number of elements of the model is finite.

We cope with this problem by demanding that each element of our model be able to reach a distinguished state, called dormant state, whose meaning is "no information is carried by the corresponding element at this time". Then, we satisfy the finiteness requirement by decreeing that at any time, all but a finite number of elements in the model are in dormant state. Thus, despite the fact that there is an infinite number of available elements in the model, only finitely many of them are in a state that carries nontrivial information.

In Chapter 4 we will see that the dormant state, in addition to

the role described above, has the function of separating simultaneous non-interacting epigenetic control processes in the informational space.

We conclude this section with a brief introduction of a special element in our model which simplifies our analysis.

We have seen in Section 2.1 that the model consists of ensembles of elements classified[1] according to their operational role in epigenetic control processes, and in this section we established that these elements are allocated to points of the informational space. From the analytic point of view, it is very difficult to determine properties of such a model.

We cope with this difficulty by constructing an element capable of simulating, in a precise sense to be given in Chapter 4, the behavior of all the classes of elements in the model (e.g. elements representing operons, MRNA's, proteins, transcription feedback mechanisms, etc). We call this element the covering element.

Thus, if we assign to each point in the informational space a covering element, by carefully setting it so as to simulate the appropriate element, we will have the same behavior in this model as we would have in the model with the original elements; with the advantage that now we are dealing with a single element at each point in the informa-

[1] In Section 2.1 we describe this classification informally. A formal description is given in Chapter 3.

tional space.

2.3 Scope and Aims of the Study, Fundamental Assumptions and Experimental Data Base

In this section we establish the objectives of our study, state the formal procedure we follow to achieve these objectives and discuss its limitations and the simplifying assumptions, about the operational characteristics of the genetic apparatus, made throughout the development of this formal procedure.

Our main objective is to develop an algebraic model that captures the important operational features of the epigenetic control mechansims introduced in Section 2.1, and on the basis of this model study the dynamic characteristics of 3 types of epigenetic control processes in procaryotes. These are

- Initial Activation Processes
- Composite Processes
- Construction Processes

A brief description of these processes is given next.

By initial activation processes we mean an epigenetic process which starts with the activation[1] of one or more operons at a given initial time t=0, say. The process then evolves as in the example of Figure 2.1-1 without any further operon activations.

---

[1] The corresponding elements representing the operons go from off condition to on condition (see Figure 2.1-5).

Composite processes are epigenetic control processes which during their evolution interact at specific times with a main stream control process (perhaps a better description would be that those processes merge at specific times with the main stream process). We will see that the characterization of these processes can be accomplished naturally in terms of our model. (see Section 4.5.2).

Each of the processes forming a composite process is an initial activation or composite process. In composite processes, the classification of processes according to hierarchies mentioned in Section 2.1 plays a crucial role in the analysis of process coordination, that is, how the information of interaction times among processes is contained in the processes themselves.

A construction process is an epigenetic control process whose function is to construct the initial state of an initial activation process which then evolves independently of the process that carried out the construction.

By the initial state of an initial activation process we mean the states of each of the elements in a scenario such as the one digramatically illustrated in Figure 2.1-7.

Construction processes are not only important for the role they play in the genetic apparatus, but as we shall see in Sections 4.5 and 4.6, they constitute a very important analytic and computational tool for the study of composite processes.

In general, an epigenetic control process in procaryotes involves processes of the 3 classes described above; thus, the study of these prototypes of control processes constitutes a preliminary step on the basis of which an effective methodology for the analysis of epigenetic control processes can be built. We shall illustrate this assertion in Chapters 3 and 5 with some examples.

The fundamental task in our main objective is to develop a mathematical representation for the dynamics of our model. Since the model dynamics is determined completely by the dynamics of its elements, we must find a representation for them.

On several occasions in the last two sections we have referred informally to the state[1] of the elements as the object that determines its information-content at each time. It is therefore natural to express the dynamics of an element by the evolution of its state with respect to time.

In this study, the state of every element in the model evolves in discrete, equally spaced time intervals which, without loss of generality, we take as one unit in length. This, of course, constitutes an approximation to the true nature of time in the dynamics of epigenetic control processes. We will examine its consequences later in this section. For the moment, we remark that time in our model is a

[1] A formal description of the state of the elements is given in Chapters 3 and 4. In Chapter 3 it is defined from an operational point of view. In Chapter 4 a mathematical definition is given.

variable taking values in the non-negative integers.

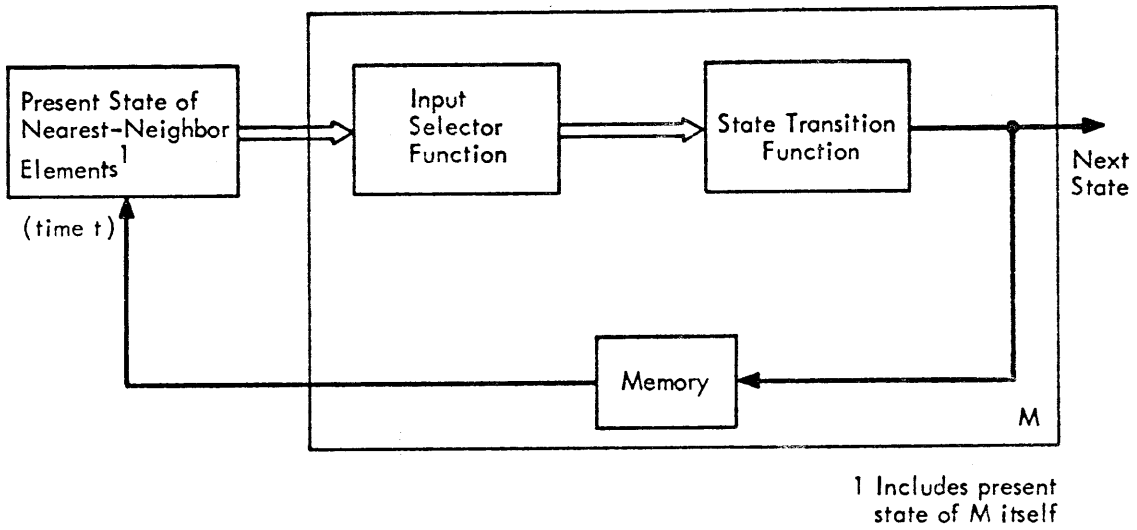
The evolution in time of an element M is characterized by two functions. A function representing the input selector described in Section 2.2 and a function, which given the states of the selected nearest neighbor elements of M at time t, gives the state of the element M at time t+1. This function is called the state transition function.

The fundamental task in our modeling effort consists in finding the transition function for each of the distinct types of elements that participate in the 3 classes of epigenetic control processes defined above, and from those, constructing the transition for the covering element. This is carried out from an operational point of view in Chapter 3 and then, in Chapter 4, a mathematical representation of this function is proposed.

Figure 1 illustrates with a block diagram the general structure of an element in our model. We note that in addition to the Input selector and state transition functions, the element possesses a memory to store the state for one unit of time.

Now we mention a second simplifying assumption with respect to the state of the elements of the model and assess its effects in our study. We will assume that the number of different states that the element of our model can attain is finite.

In principle, the finiteness of the number of states an element may attain, can be justified from physical considerations, since the ele-



[1]Includes present state of M itself

General Block Diagram Representation of
One-step (in time) State Transition

Figure 1

ments of the model represent molecules, (or actions carried out by molecules) and it is accepted that molecules as quantum-mechanical systems can exhibit only a finite number of energetic states (the discrete eigenvectors of their Hamiltonian operator or combinations of them (pure states and mixed states respectively)). (see Calvert and Pitts [51]).

A typical organic molecule has several thousands of distinguishable energetic states, and their transition structure determines its functional characteristics. However, the operational status of the element corresponding to relatively large subsets of energetic states is very similar (i.e., a molecule in any of the many vibrational ground states operates in essentially the same manner). Therefore, since the concern of this study is with operational aspects of these molecules, our element's states correspond to aggregated[1] classes of energetic states each one representing a functional status of the element.

The simplification discussed in the last paragraph makes the modeling problem tractable (i.e., the number of different states to be considered is reduced by 2-3 orders of magnitude (typically several thousand states are aggregated into several states) at the expense of losing some operational detail in the dynamical characteristics of the elements. Unfortunately, since our strategy for constructing the operational structure of the elements in the model does not include

---

[1]This state aggregation is not homomorphic as opposed to the informational state aggregation to be introduced in Section 3.4 in which the dynamics is preserved.

determining their energetic state structure, we cannot give a formal procedure for evaluating the severity of our simplification.

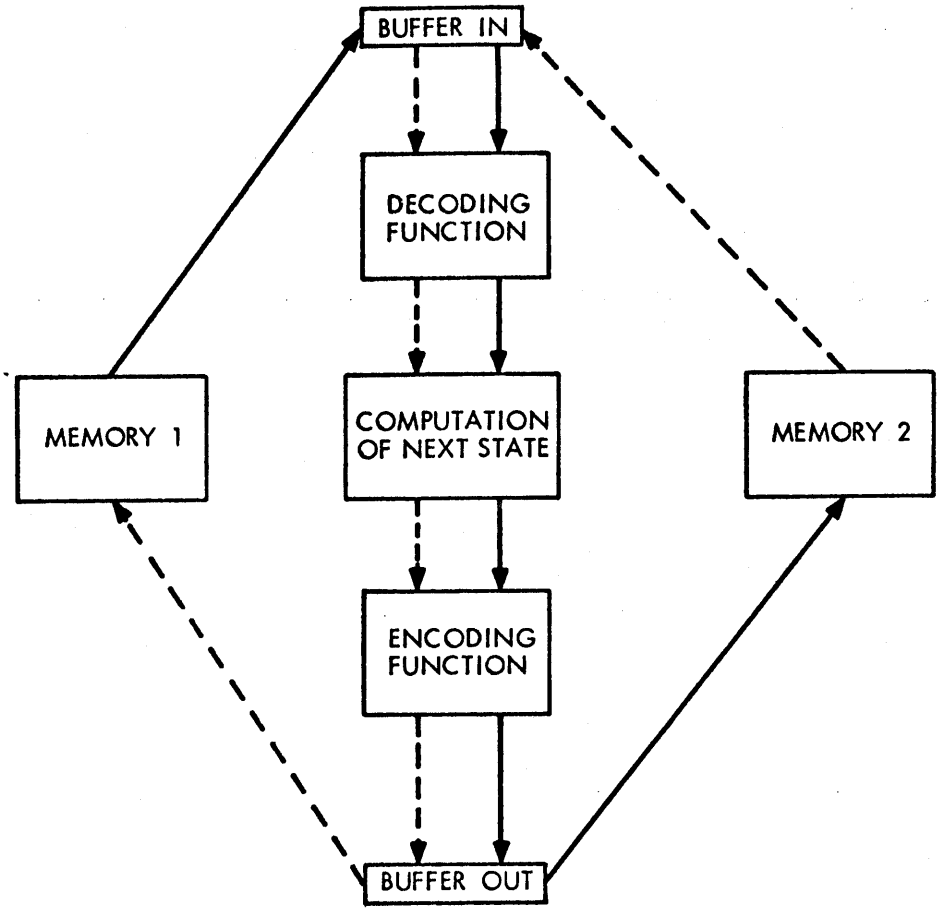
In recognizing the importance of at least testing the limitation of our model with respect to the simplifying aggregation procedure introduced above, as well as other simplifying assumptions, we resort to a heuristic procedure which involves a computer simulation program of our model and a structural identification procedure on the basis of expreimental data. Specifically, we will analyze several well-studied biological examples to see if our model predicts behavior that is known to be true from experimentation. These two aspects are discussed next.

The basic structure of the computer simulation program is shown in Figure 2. The program consists of 7 functional blocks; two independently addressable storage areas labeled buffer in and buffer out; and 3 computation blocks labeled decoding function, computation of next state and encoding function.

At any time t of the simulation, and encoded[1] version of the state of each element is stored in one of the memories, in an address which is a function of its coordinates in the informational space. Only elements in non-dormant states are stored.

For the purposes of illustration, let us assume that at time t, the states of the non-dormant elements are stored in Memory 1. The

[1]The encoding is discussed in Chapter 5.



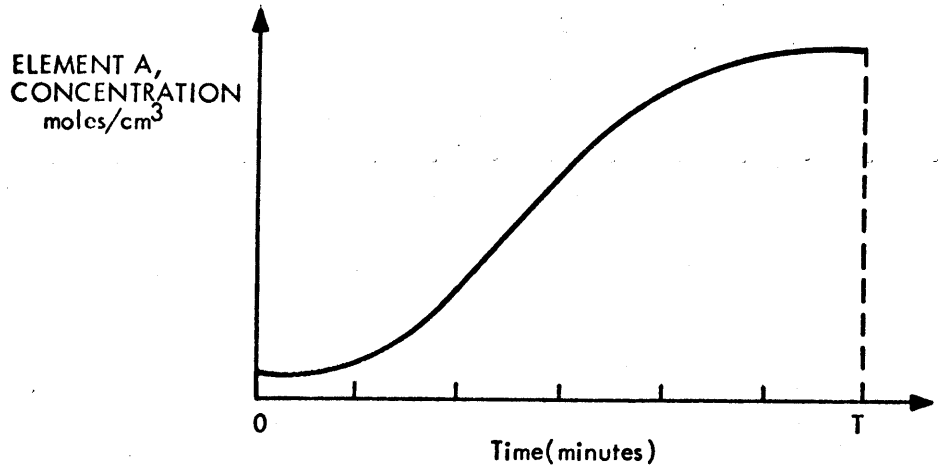
Conceptual Block Diagram of Computer
Implementation of the Model

Figure 2

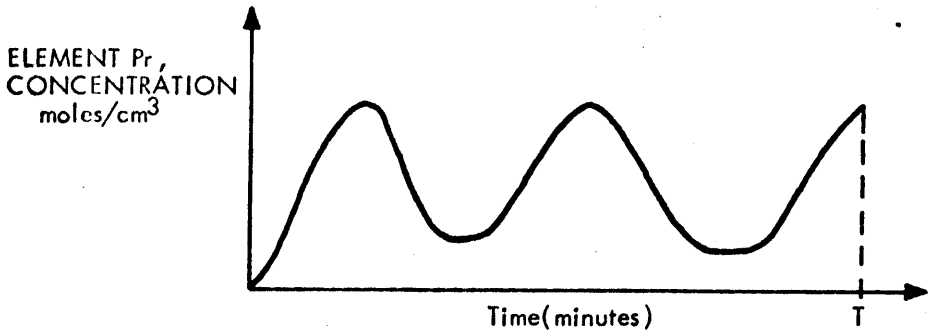
simulation proceeds by bringing to buffer 1, in a preassigned order, the encoded version of the states of a group of nearest-neighbor elements (2k+1 elements see Section 2.2), of a given element M, they are decoded, and the state of M corresponding to t+1 is computed, encoded and then placed in the buffer out. Next, the same procedure is repeated for another element until the buffer out capacity is exhausted; then its contents are transferred to Memory 2. This procedure is repeated until all the states of the non-dormant elements at time t+1 are computed. This completes the simulation of one step (in time) of the model, (referred to in the figure as 1-2 cycle).

Now, we have the states of the non-dormant elements corresponding to t+1 in Memory 2 and we are ready to compute the t+2-states and store them in Memory 1 (2-1 cycle), following the same procedure outlined above. Several technical aspects of this procedure such as parallelism of the simulation to reduce computation time and the encoding and decoding operations are discussed in detail in Chapter 5; but the basic characteristics of the simulation are implicit in the diagram of Figure 2.

Now we discuss the structural identification procedure. The first most important ingredient in this procedure is the experimental data base and how it is presented. Let us assume that we are studying an epigenetic control process in which we have determined by any of several possible experimental techniques [50], the identity of the n classes of molecules that participate in its reaction systems and that from those



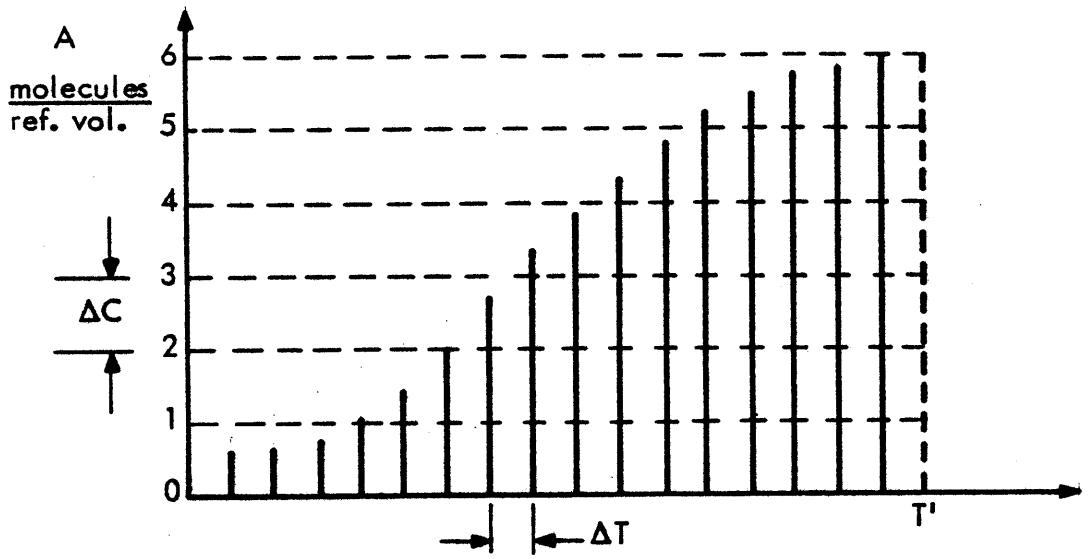
(a) Time course concentration of element A



(b) Time course concentration of Pr

Experimental Data

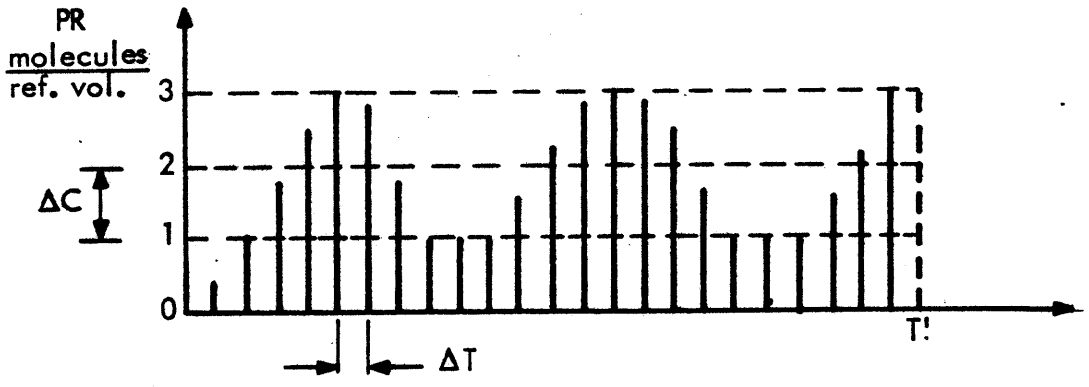
Figure 3



(a)

ΔT Time discretization interval
= 1 unit in terms of the time
scale of the model

ΔC Quantization interval
= 1 unit



(b)

Illustration of Quantization and Discretization Operations
on Experimental Data of Figure 3

Figure 4

m, ($m<n$) are proteins. Further, let us assume that the data about the process is given as time-course concentrations for each of the n molecules with respect to a basic volume (that we always assume is the intra-cellular volume, a constant). Figure 3 gives two examples of the way this data is presented (i.e., as concentration vs. time graphs).

As a first step in the processing data, the concentration vs. time graphs are transformed into a new set of graphs in which the time axis is a discretized version of the original time axis and the concentration axis is a quantized version of the original concentration axis. Figure 4 illustrates these two operations for the examples of Figure 3.

The interval (time) of discretization, ΔT, is chosen by a criterion[1] discussed in detail in Chapter 5. We mention here that this criterion is determined as a function of the adjusted time horizon of the experiment (T' in Figure 4)[2].

Now, before discussing how the transformed data is used in the identification procedure, we need to introduce a third, very important assumption which justifies the strategy adopted for this procedure. Because of its importance, we have assigned a name to this assumption: the closure assumption. The implications of the closure assumption are

---

[1]The criterion under which ΔT is determined, is called the Nyquist sampling-rate theorem (see Oppenheim and Schaffer [52]). This theorem gives a constructive criterion for sampling a continuous function (e.g. a concentration function) with sampling interval (ΔT) sufficiently small such that the function can be reconstructed from its samples.

[2]The adjusted time horizon T' converts experimental data, usually measured over several cell generations to equivalent data over a single generation.

are analyzed from several points of view throughout this study. (In Chapters 3 and 4, in terms of the model, and Chapter 5 from the simulation point of view.) In the next paragraph we given an informal description of this assumption.

The closure assumption establishes that every epigenetic control process evolves according to a structural arrangement of elements of the type indicated in Section 2.1. See Figures 2.1.1 and 2.1.6). Further, the reaction systems of these processes are of one of the forms indicated in Figure 2.1.4 or can be decomposed into finite number of steps of these forms.

The most important implication of the closure assumption with respect to the structural identification procedure is that the epigenetic control process from which the data is a time-course observation is of this type. Therefore, since the model has been derived on the basis of these structural arrangements of elements, we should be able to reproduce the quantized and discretized version of the data with the simulation procedure previously described.

Now we return to the description of the identification procedure. The next step in the procedure is to guess the dimension of the appropriate informational space. This guess will be recursively updated during the procedure (in the actual computer implementation of this procedure the initial value of the dimension is always 3).

Once a choice of the informational space has been made, we proceed to assign elements representing the molecules present in the system

at time T' to locations of the informational space (one location for each molecule; for instance, for the molecule A in the example of Figure 4a, we assign 6 A-elements. The criterion for this assignment follows from the closure assumption plus the nearest-neighbor requirement of our model and its technical aspects are given in Sections 5.2 and 5.3 in terms of two specific examples. We note here that in order to satisfy the nearest-neighbor requirement, we may need to update the guess of the dimension of the information space.

The next step consists of finding a set of state assignments for the elements in the informational space such that any one of them, when used as the initial state assignment for the simulation algorithm outlined previously, will produce the state assignment constructed above after one step of the simulation procedure. Now, any one of the states assignment in this set is a candidate for the state assignment of the elements participating in the process at time T'-1; from these we choose one that corresponds to the experimental data at time T'-1.

Next, we repeat the procedure discussed above with the state assignment corresponding to T'-1; that is, we construct a set of state assignments of the elements such that one step of the simulation procedure will produce the state assignment corresponding to T'-1. From this set we select a state assignment that agrees with the data at T'-2.

This procedure is repeated until an assignment of state is found that corresponds to an initial activation process or to the main stream process of a composite process.

The outcome of the structural identification procesure, (if successfully completed)[1] gives us a diagram of the organization of operons and reaction systems for the process under study (i.e., a diagram of the form of the ones given in Figures 2.1.1 or 2.1.6), with its corresponding diagram of events.

Several important operations with respect to the identification procedure are in order. First, the adjusted horizon T' corresponds usually to several cycles of the epigenetic process; that is, the first operon(s) to be activated in the process, an event corresponding to time t=0 in the time basis of the model, does not necessarily (and usually does not) correspond to the initial time of the experimental data. This does not impose any restrictions on the effectiveness of the procedure because the halting condition is determined by a special class of state assignments of the non-dormant elements of the model, and not by time.

Second, we note that the experimental data is produced through measurements on the process that are always of finite precision and subject to environmental disturbances. These considerations are ignored in the formulation of the procedure outlined above, where we have made the implicit assumption that the data is error free. Fortunately, the quantization operation tends to decrease these effects (see Section 5.2).

---

[1]Since the structural identification procedure is not an algorithm it may not terminate for some tasks.

Third, in the construction of sets of feasible state assignments, that is done recursively in the procedure, we have found that this task can be effectively accomplished if we use a representation of the model as a set of polynomial equations. This representation of the model is developed in Section 4.6.

The effectiveness of the polynomial representation of the model with respect to the identification procedure, lies in the fact that an algorithm for computing the set of feasible state assignments at some time t given the state assignment at t+1, can be implemented in this representation as the procedure for finding the solution of a set of polynomial equations. This algorithm is developed in Section 4.6.

It is important to note here that the set of equations corresponding to a given step t in the identification need not have a solution. If this is the case, it means that there is an error in the experimental data in the time interval [t+1, T'] or, the data corresponds to a process in the genetic apparatus whose operational structure (i.e., time-space interaction of the elements participating in it) has not been incorporated into the structures from which the model was derived (see Section 2.1). In Sections 4.6.1 and 4.6.2 we develop generic solvability tests for a class of polynomial equations which contain, as special cases, the equations that appear in the identification procedure in the backwards (in time) iterative construction of state assignments.

The product of the identification procedure is either the structure of the corresponding epigenetic control process, which is presented

as an incidence-element diagram such as the ones displayed in Figures 2.1.1 and 2.1.6 together with the corresponding time-event diagram such as the one given in Figure 2.1.3, or,"no structure that satisfies the experimental data could be found". In both instances the procedure will be shown to terminate in finite time (see Chapter 5).

We now leave this outline of the identification procedure and conclude this section with a brief discussion of several additional assumptions and simplifications made in the development of the model.

We note that with respect to time our model has synchronous dynamics, that is, a state transition of all the elements in the informational space, occurs simultaneously. At first sight, this characteristic does not seem to correspond to the operational characteristics of molecules in the cell whose collective dynamics is asynchronous with local synchronous behavior only among molecules that interact with each other. However, the synchronous behavior of the model is not a limitation in its capabilities of simulating epigenetic control processes because we can introduce asynchrony in these processes by means of the virtual elements in the model labeled clocks. (see Section 2.1). Thus, for example, in simulating the reaction system,

$$A + B \xrightarrow[PR_1]{} C + D$$

$$C + E \xrightarrow[PR_2]{} F$$

We will simulate the following

$$A + B + W_1 \xrightarrow[PR_1]{} C + D$$

$$C + E + W_2 \xrightarrow[PR_2]{} F$$

where W_1, W_2 provide the appropriate time scaling between the two reactions. Other instances of asynchrony are treated similarly.

Finally, we mention that our model assumes that the epigenetic apparatus operates in error free manner that is mis-sense mutations in the operons, errors in transcription or protein synthesis which are phenomena that occur in the apparatus are not considered in our model. We believe that for our purposes this is not a serious limitation because our main interest centers in using the model for studying wild type operons and not their mutation mechanisms which seem to be of stochastic nature, (see Holland [22]).

2.4 Summary

In this chapter we have described our goals in the study and outlined the strategy we follow in accomplishing them.

In summary these goals are:

-- Development of a model which captures some important operational aspects of the dynamics of epigenetic control mechanisms in the genetic apparatus of procaryotes, and that can be used to predict operational behavior in their epogenetic control processes.

-- Analysis of 3 special types of epigenetic control processes in terms of the model. These are:

- Initial Activation Processes
- Composite Processes
- Construction Processes

-- Development of a structural identification procedure whose objectives are to provide the means for identifying some structural characteristics of real epigenetic control processes from the operational point of view and also, to test the capabilities of the model as a research tool in genetics of procaryotes.

The most important characteristics of the model are:

-- The model is an ensemble of elements each one of them representing the functionality of a molecule or action in the genetic apparatus.

-- These elements are distributed in a k-dimensional grid and the dynamics of each element is characterized by its state transition in time as a function of the state of one or all of its nearest-neighbors in the grid.

-- The time basis of the model is discrete and each of its elements can exhibit only a finite number of distinct states.

-- At any time, only finitely many elements in the model are in a useful state from the operational point of view.

3. BASE MODEL FOR EPIGENETIC CONTROL MECHANISMS IN PROCARYOTES

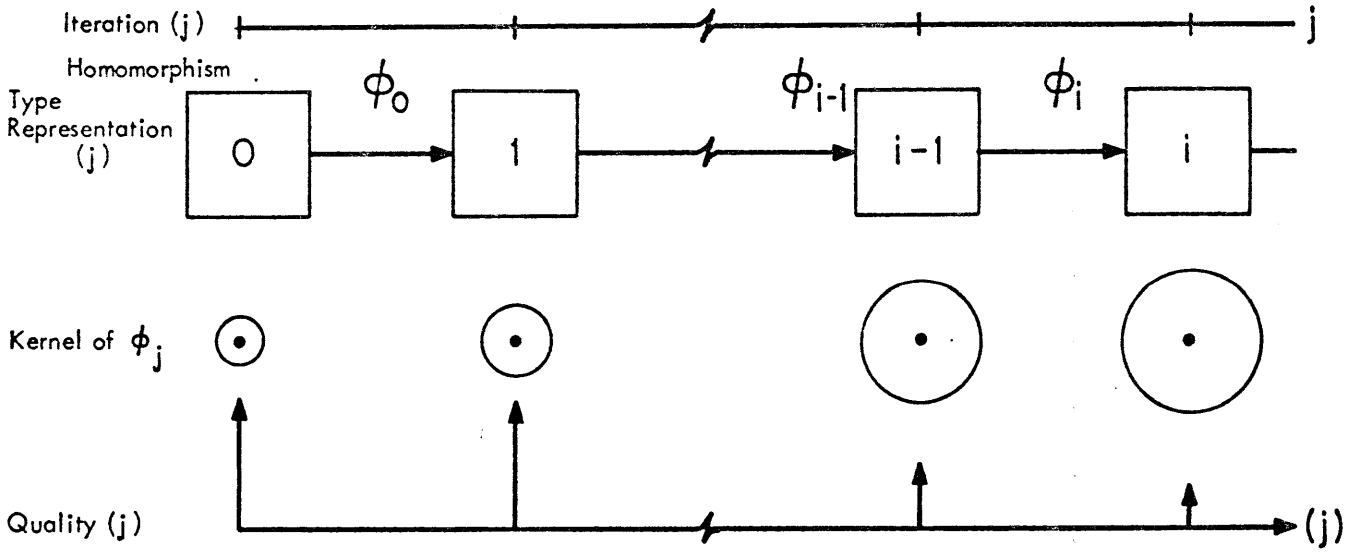
3.1 Outline of Modeling Strategy

In this chapter, we describe in detail the main operational characteristics of our model, and establish the relation that exists between them and the corresponding features of the genetic apparatus.

The outcome of this chapter is a more or less detailed verbal description of the structure of the model and its dynamic character. Following usual notation in modeling theory, (see Ziegler [16]) we call this description the base model.

The modeling procedure is composed of the following four steps:

1. Classification of the components of the system according to their function in epigenetic control processes.
2. Development of a typical representation for each class of components. The resulting model-elements are called the types of the system or simply types.
3. Development of the incidence-diagram of dynamic interaction among elements of the classes. This diagram establishes for each type, the types of the classes with which it interacts, and the direction of this interaction. The diagram is constructed on the basis of an analysis of the functional interaction of the corresponding elements of the system.
4. Aggregation of the descriptive variables of the types (i.e.,



Quality (j) = (cardinality of Kernel of ϕ_{j-1})$^{-1}$

Illustration of the Iterative Homomorphic Aggregation of the Types of the Model

Figure 1

the variables representing their states) in order to obtain a simplified description of the model. In this step, we obtain a computational representation of the types at the expense of loss of detail in the description of processes in the genetic apparatus by the model. In out study, we use a technique for aggregation called homomorphic aggregation (see Aoki [1]); in it, the quality of the resulting representation is measured by the cardinality of the kernel of the aggregation homomorphism which is described in Section 3.4.

Our goal is to obtain a model that can be used as a general tool for investigating epigenetic control processes in procaryotes. In particular, we expect our model to provide an effective procedure for setting out structural hypothesis about the epigenetic mechanisms responsible for observed experimental data (i.e., structural identification of these mechanisms, see Chapter 2). Therefore, we must provide the procedure above, with a routine for producing an aggregated model which is unambiguous (i.e., well posed) with respect to the given experimental data and that is as simple as possible from a computational point of view.

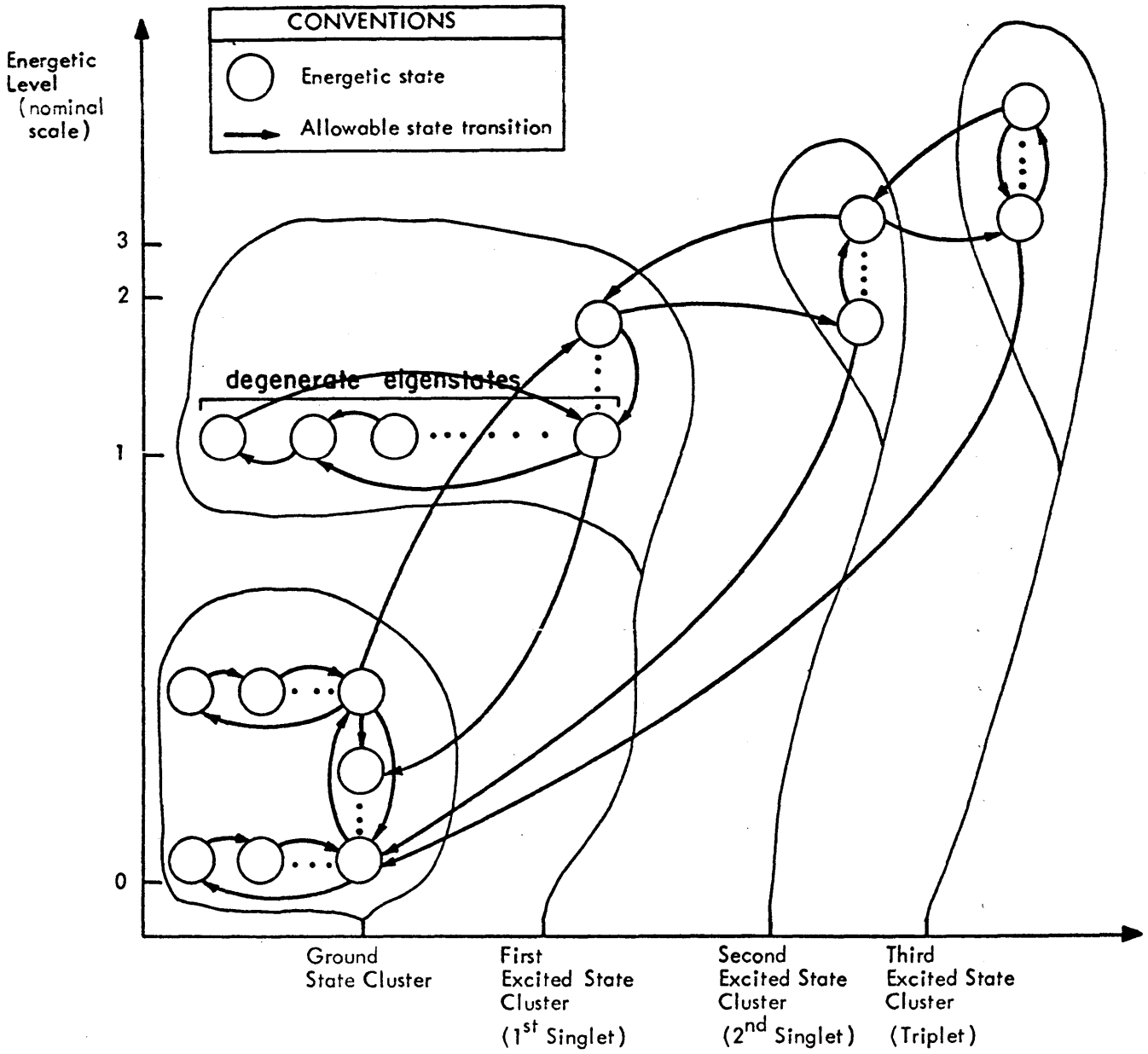
With this objective in mind, we devise an iterative scheme for the fourth step in the procedure. In it, each iteration computes an aggregated set of types whose quality is lower than that of the previous iteration (i.e., cardinality of the kernel of the corresponding homomorphism is higher that the one corresponding to the previous iteration).

This is illustrated with the diagram of Figure 1.

We note that by definition, homomorphisms preserve structure, in our case, the structure of types. As mentioned in Section 2.3, the structure of the elements of the model and consequently, the structure of the types, is given by the one-step (in time) state transition function of the element (see Figure 2.3.1). In the remainder of this section we introduce this function as well as the general structure of the state of an element and justify it from a practical point of view. It is safe to say that this function has determined in great part our modeling strategy.

We start with a description of the state structure of an element. Recall that an element is a representation of a molecule or an action in the genetic apparatus, therefore, we analyze the operational characteristics of the dynamic evolution of molecules to determine faithful generic properties of this evolution. With this purpose in mind, we digress briefly to outline the state structure of a typical organic molecule from a microkinetic point of view (see Section 2.1). This will give us the basis for the development of the operational state structure of our model.

Figure 2 is a representation of the energetic state structure of a typical organic molecule. The cirlces correspond to the eigenstates of the Hamiltonian operator associated with the molecule. Each eigenstate is associated with a unique energetic level, the eigenvalue of the operator (discrete spectrum)but the converse is not true in general



Energetic State Representation of a Typical Organic Molecule

Figure 2

(this is called energetic degeneracy in the quantum chemistry literature, see Hanna [2]); that is, with a given eigenvalue there are, in general, several eigenstates each corresponding to a unique arrangement of electron orbits, electron suborbits and electron spin orbits in the molecule. (This is called the Pauli exclusion principle; see Hanna [2].)

The state degeneracy alluded above has very important consequences in the functional behavior of the molecule: i.e., a condition of <u>state transition indeterminancy</u> from spectroscopic observations of the energetic structure of the molecule, a fact that is reflected in the non-unique operational behavior of the molecule's model when we aggregate energetically equivalent states.

The arrows joining the states represent <u>allowable</u> state transitions. Each of these transitions is driven by energetic interactions with elements in the near environment of the molecule.[1] We will not go into the details of the mechanisms of interaction because they are not needed for our purposes; it suffices to say that these mechanisms have been extensively studied during the last 50 years and although some of their aspects are not yet understood, the basic nature of these interactions is known: they always involve the transfer of energy between the participating molecules. See for example Calvert and Pitts [3], Byam [18], Turro [4], Dickerson [3], Hanna [2], etc.

---

[1]The interactions may be of several kinds, e.g., electromagnetic radiation, chemical bond formation, transfer of molecular residues, etc. But always, their effect can be represented as a flow of energy to or from the molecule, or between groups of the molecule (radiation-less transitions).

As indicated in Figure 2, the energetic states are grouped into several clusters each one of them corresponding to an energetically observable condition of the molecule (from a spectroscopic point of view). The ground state cluster, for instance, corresponds to the situation of lowest energy stored in the molecule.[1] The cluster identified as first exited state cluster (or lst singlet in specrtoscopic literature) corresponds to the first exited group of states the molecule can obtain when the environment provides the appropriate transition energy, and so on.

Each cluster is composed of several states (in some molecules several hundreds of them) whose energetic level differences are small as compared with the intercluster energy levels. These states represent different degrees of thermal exitation of the molecule, and transitions among them are induced by small random local temperature gradients or vibrations of the bounds of the molecular structure (see Turro [4]).

From an operational point of view, the differentiation between thermal states in a cluster is only secondary[2] and therefore, in what follows, we will aggregate them into a single state in such a manner that inter-cluster state transitions are preserved. In this aggregation procedure, inter-clusted transitions are preserved provided that in each cluster, the vibrational states are completely connected; that is,

---

[1] By "energy stored" in the molecule we understand the energy capable of doing work, that is the excess over the energy required by the molecule to maintain a stable morphogenetic structure.

[2] However, we will use that state degeneracy alluded before to justify a state transition indeterminancy that appears in the transition function of the types of the model.

any state of the cluster can be reached from any other state in it by an appropriate (finite) sequence of allowable state transitions. Fortunately this is the case in most of the organic molecules on which spectroscopic analysis have been carried out.

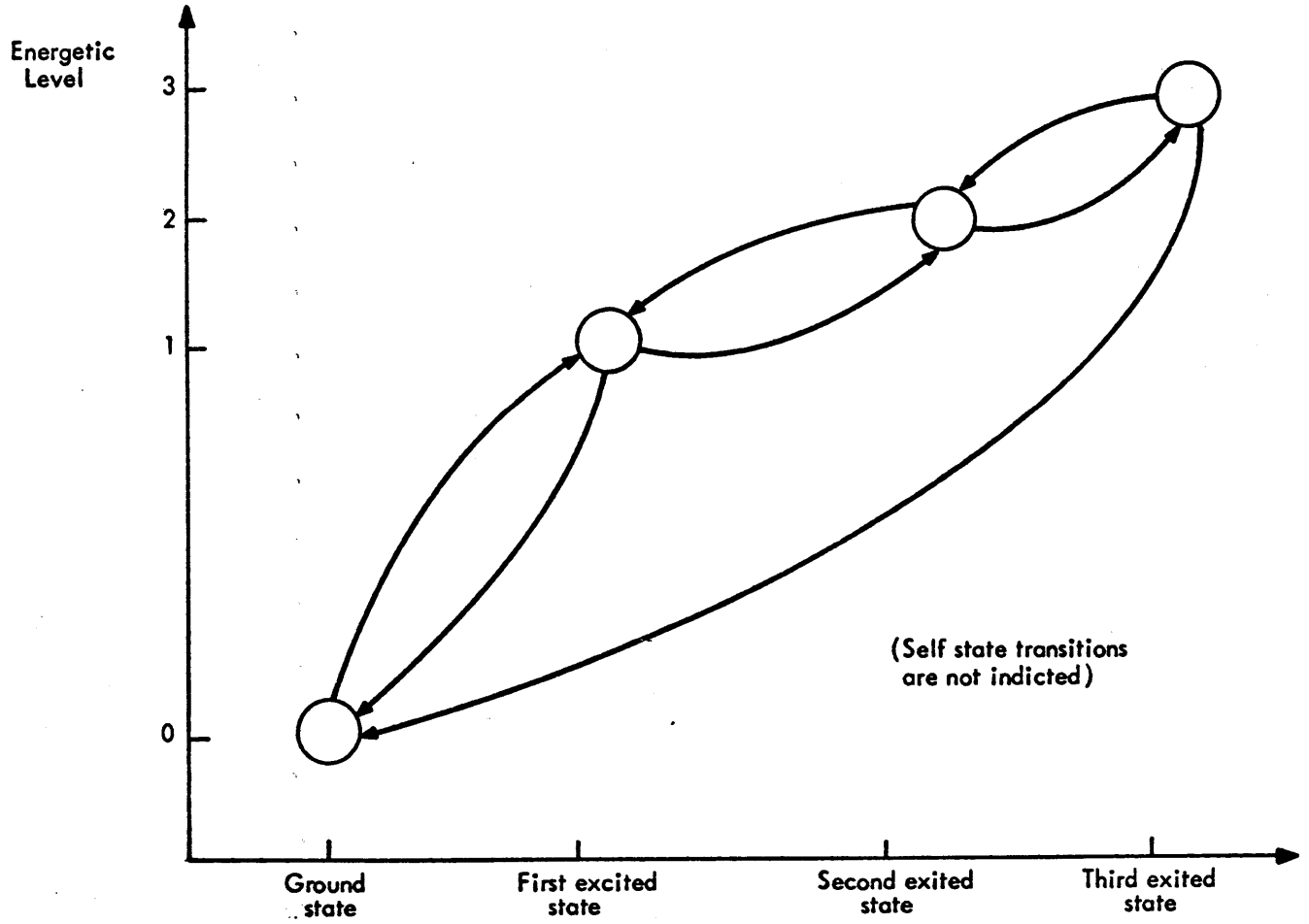
If the vibrational states in a cluster are not completely connected, we divide it into sub-clusters whose states are completely connected and then proceed to the aggregation procedure.

Under the aggregation procedure defined above, the state diagram shown in Figure 2 takes the form of Figure 3. This leads to a non-deterministic state transition diagram. However, any nondeterministic automaton can be simulated by a deterministic one (see Ginsberg [6]), and thus in Chapter 4 we can define our mathematical framework using only deterministic automata.

Now, we consider the interaction between two or more molecules in terms of the interaction of their corresponding energetic state diagrams.

The dynamics of an ensemble of interacting molecules from the energetic point of view, can be developed, in principle, following the same pattern as the one outlined above for a single molecule; that is, we find the Hamiltonian operator of the ensemble, determine its discrete spectrum with associated eigenstates (the energetic states of the ensemble) and finally, determine the allowable state transitions.

It is well known (see Hanna [ ]) that this procedure fails from a practical point of view due to the impossibility of implementing an



Aggregated Energetic State Representation of a Typical Organic Molecule

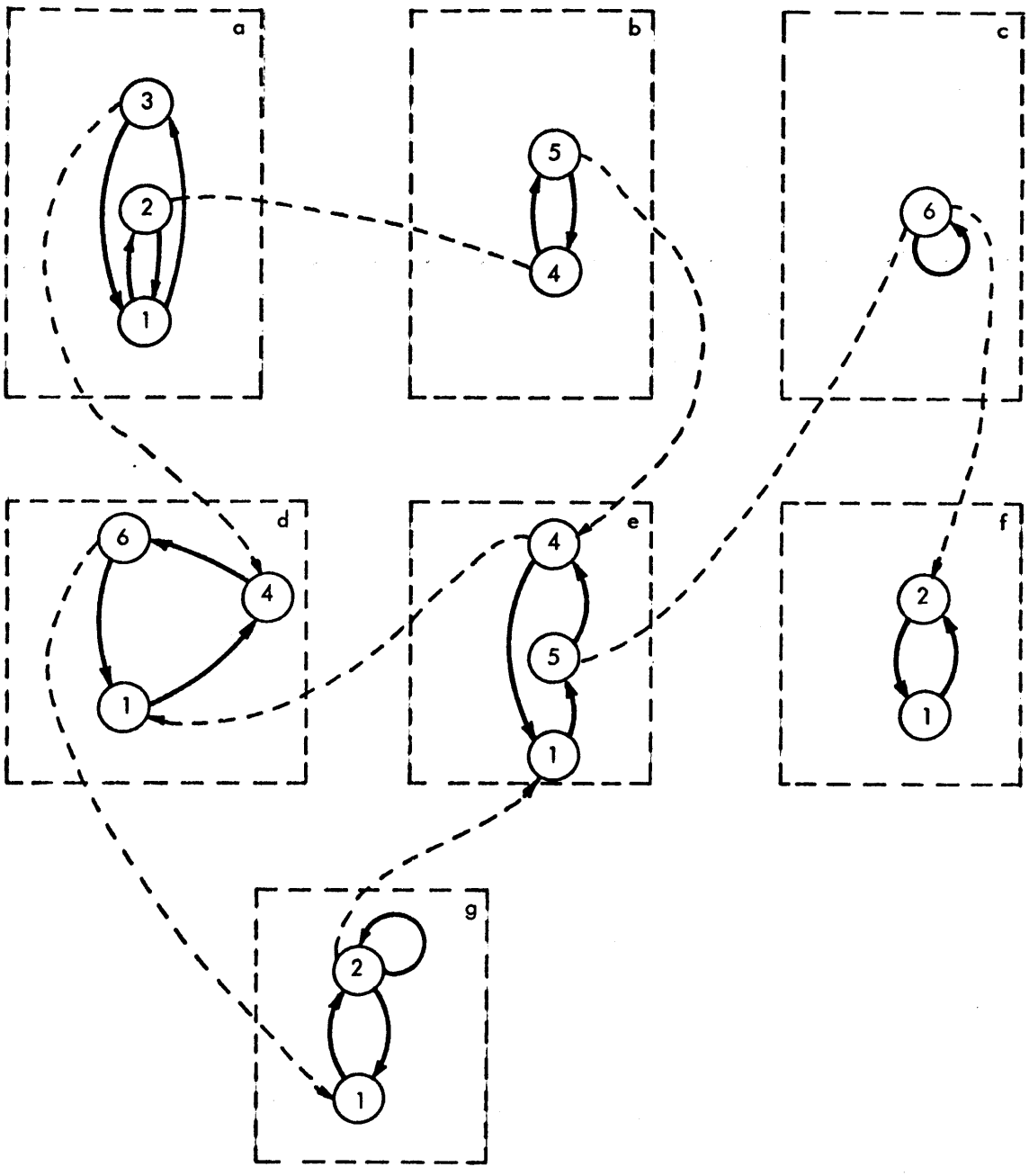
Figure 3

effective computation of the spectrum of the Hamiltonian. Therefore, in practical spectroscopy, researchers have resorted to several approximations, among them, the most popular are: perturbation-based approximations (see for instance, Byam [18]) and heuristic-based approximations among which, the most successful one is Dendral which is built as a recursive learning model (see Buchanan et al [7]). We will introduce next a third approximation, based on some general functional considerations about the energetic behavior of the interacting molecules in the ensemble. This approximation, constitutes the cornerstone in the development of the state structure for the types of our model.

The alluded approximation is based on two postulates:

-- A typical organic molecule has a highly selective chemical activity; that is, there are only a finite number of reaction systems (see Chapter 2) in which the molecule participates. This is translated into the fact that its energetic structure is not affected by most of the other molecules that may be present in its near environment. (An important exception to this behavior is observed in Immunoglobulin and several other macro-molecules with which we are not concerned in this study (see Roitt [8]).

-- When a molecule (or action) does interact with other molecules in its near environment and there is chemical affinity between them, the interaction Hamiltonian (i.e., the Hamiltonian of the ensemble)



Interaction State Diagrams of a typical molecule --

Circles indicate energetic state, dashed rectangles indicate the different environment

Figure 4

is a function (in general non-linear) of the Hamiltonians of the individual molecules.[1]

The two postulates above, imply that the state diagram of a molecule at a given time depends on the molecules in its near environment and also the number of different state diagrams attained by the system formed by the molecule and the different environments with which it interacts, is finite. This assertion follows from the fact that if the molecule does not interact with a particular environment, its Hamiltonian and its state do not change.

This suggests the following strategy for representing interacting molecules from the energetic point of view:

-- Associate with each molecule a finite number of energetic state diagrams each one corresponding to a molecular environment which is chemically affine to the molecule (this is illustrated in Figure 4).

-- Determine the allowable transitions between the different state diagrams of the molecule (dashed arrows in Figure 4). These transitions are determined from a spectral analysis of the interaction Hamiltonian. We will not carry out this analysis here because it is not needed for our purposes, which are to develop a state transition function for the elements of our model. However, we believe that the

---

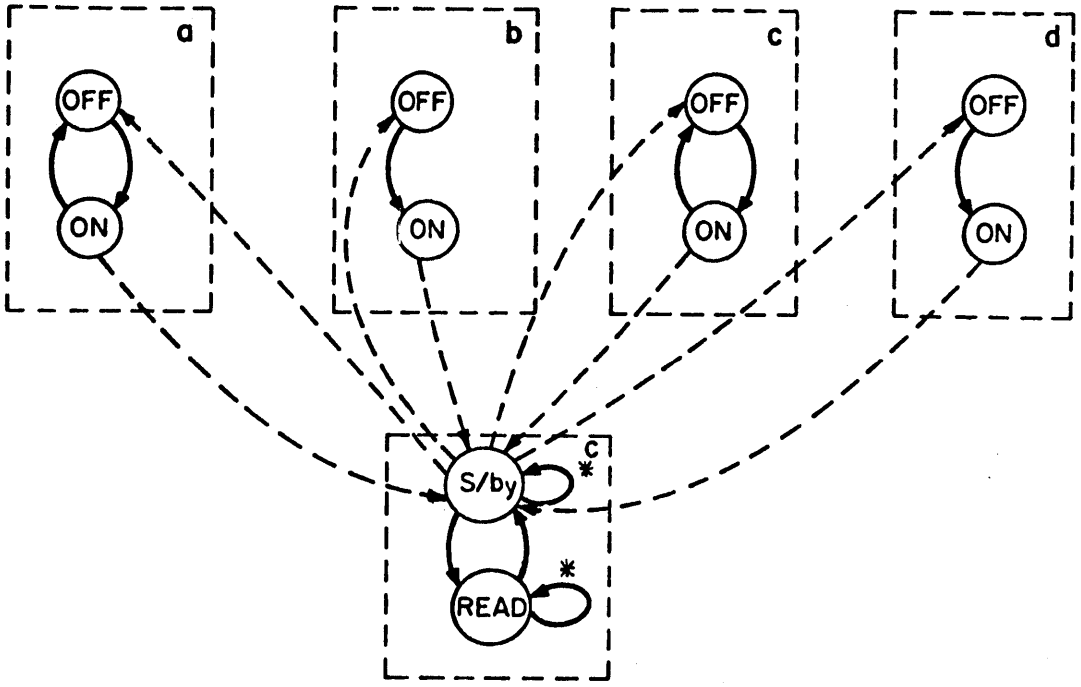
[1]Let H_o be the Hamiltonian of the molecule in question and let $H_1,\ldots,H_\ell$ be the Hamiltonians of the molecules in its environmment. $H_o,\ldots,H_\ell$ are operators in a normed vector space V. Then the interaction Hamiltonian is given by a function $f:V^{\ell+1} \to V$ $(H_o,\ldots,H_\ell)\, f=H$ (see Baym [ ]) f is continuous.

structure outlined above has very promising computational characteristics and can be used at an advantage over the other approximations mentioned earlier, as a model for microkinetic studies. We will discuss some aspects of this application in the conclusions of this study. For the moment, we proceed to establish, in general form, the operational state transition structure of the types of our model on the basis of the energetic state structure described above. As a first observation, we note that any operational status of a molecule is an energetic manifestation.

We give an example to illustrate this fact. An element representing an operon must be capable of simulating each of the four possible transcriptional arrangements discussed in Section 2.1 (in particular see Figure 2.1.5). In any of these arrangements, the corresponding operon, and consequently the element of the model representing operons, can be in high energy status or low energy status depending on whether the operon is on or off as discussed in Section 2.1.

Further, the ensemble formed by an operon element (OP_i in the notation of Section 2.1), a transcription feedback (TFM_i), a regulatory protein (P_i) and a control metabolite (M_i), operates according to precisely one of the four arrangements mentioned above.[1] Therefore, as in the energetic representation, we provide the element OP_i with four

[1]Operons with multiple controls are represented in our model as separate operons, each one with a single transcriptional control.



a-d correspond to figures 2.1-5a to 2.1-5d respectively. * Indicates that the corresponding arrows actually go to other intensities not involved in the activation of operons which we choose not to include here for the sake of clarity of the figure (see figure 3.3-1)

Partial operational state representation of an operon

Dashed rectangles represent structures, circles represent intensities, arrows represent allowable structure (dashed line) and intensity transitions (continuous line)

Figure 5

different transition structures, one for each of the possible arrangements.

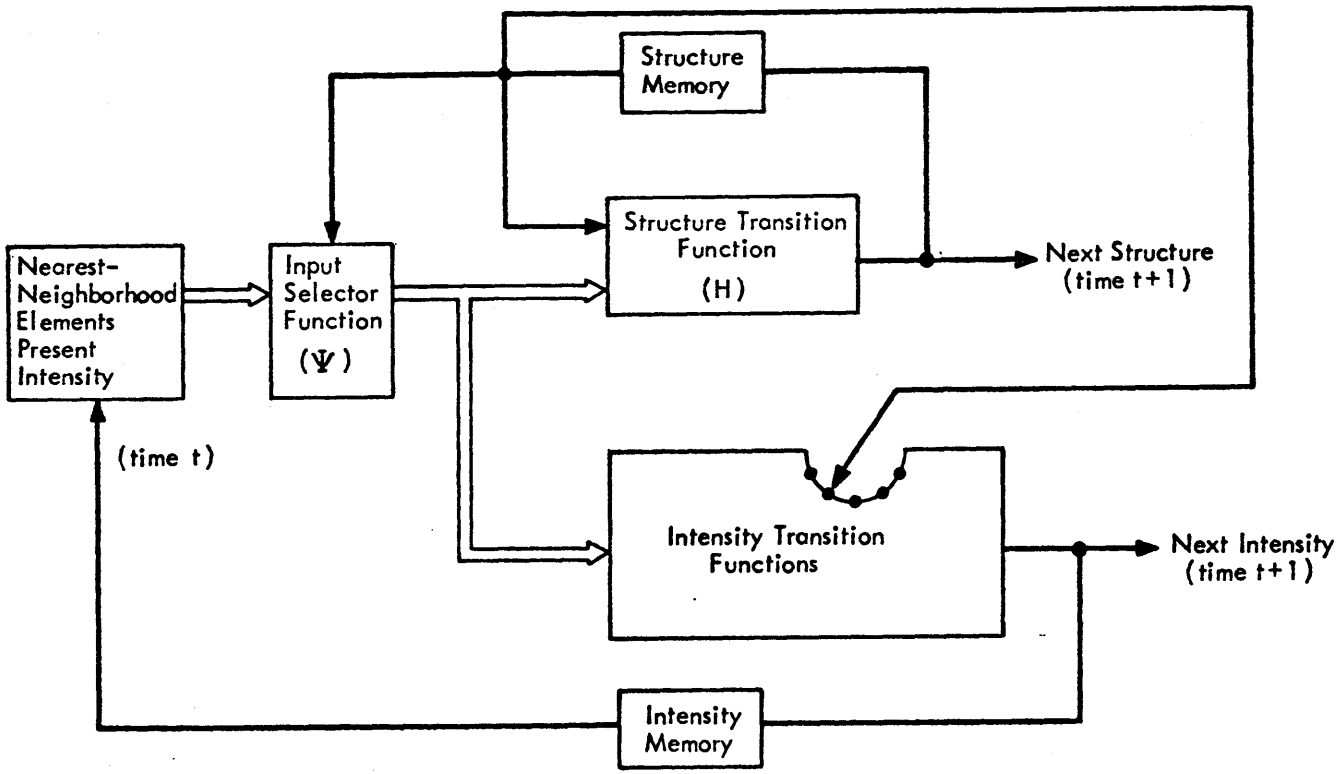
Finally, an operon element, once in ON condition transfers either to a stand/by condition or to a reading condition depending on whether an RNA polimerase is in its environment or not.

Based on the discussion above, an in analogy with the state structure outlined previously for the energetic representation, we propose as an (partial)[1] operational state structure for the operon the one shown in Figure 5. As in the case of energetic state-representation of molecules, we have two different entities in the diagram. The rectangles which represent the different structures...i.e., the nearest environments with which the element (in this case an operon) interacts. The different operational levels in each structure are called the Intensities. Thus, each operational state of the element is a composite of two objects, therefore the state transition function must be decomposable into two (coupled) functions: the structure transition function (which we always denote by H) and for each structure, the intensity transition function.

We have exhibited only one example (the operon example above), in which this operational structure of the elements fits the functionality of the corresponding molecules or actions in the genetic apparatus,

---

[1]See note in Figure 5.



Operational Block Diagram Representation of An Element's One Step (in time) State Transition

Figure 6

however, in the next two sections, after classifying the elements of the model we will show that this structure is also convenient for representing other elements and we will develop the specific representation for the types of each class.

We now establish the domains and ranges of the structure transition function and the intensity transition functions.

We recall that the time base chosen for our model is the non-negative integers and that the state transition function of an element, computes, given the states of a selected[1] subset of nearest neighbors of the element at time t, the state of the element at time t+1.

In the state representation introduced above, the set of distinct states the element can attain, is a subset[2] of the cartesian product of the sets of structures and intensities of the element.

In the energetic-state representation of a molecule a state transition is a manifestation of transfer of energy between the molecule and its near environment. In this representation the flow of energy is characterized in the element by a change in energetic level, and possibly change in structure.

In the operational-state representation, to be introduced next, a state transition is a manifestation of information-flow between a molecule or an element representing an action, and its near-environment.

---

[1] See section 2.3.

[2] We note that not all the pairs of structures and intensities are states of the element, as can be easily inferred from the example of Figure 5.

Physically, this information flow is a consequence of energy-exchange, therefore it is reasonable to adopt a dynamic representation of information-flow of analogous characteristics to the energetic-state representation discussed previously.

In analogy with the energetic state representation of a molecule discussed earlier, and based on the observation that the evolution of the dynamics of a molecule, from a functional standpoint, always involves flow of energy between the molecule and its near environment, we propose the following generic representation for the state structure of the elements of our model.

The operational representation of an element will be characterized by a transition in time of structure and/or intensity triggered by the information flowing from the nearest neighbors of the element. This information is carried by the intensities of those elements.

The representation of the state structure of an element suggested above, takes the form shown in Figure 6. Figure 6 shows that the next structure attained by the element is determined by the present structure and the selected present nearest-neighbor subset of intensities[1]. Similarly, the next intensity of the element is determined by the present structure and the selected present nearest-neighbor subset of intensi-

[1]We recall from Section 2.3 (see Figure 2.3.1) that the selection of the subset of nearest neighbors interaction with an element at any time is carried instantaneously by the input selector function.

ties.

In synthesis, the structure transition function H has as its domain the cartesian product of the set of structures and the set of all sequences of intensities of length at the most 2k+1, where k is the dimension of the informational space, and as its range the set of structures.

Similarly, we define the domain of the intensity transition functions (one for each structure) as the set of sequences of intensities of length at the most of 2k+1, and as its range the set of intensities.

In the next sections of this chapter, we will show that the operational state structure of the elements of our model is a convenient representation for the molecules and actions participating in processes of control in the genetic apparatus, and in Chapter 4 an algebraic representation of this structure is developed and in terms of it, the three (3) prototypes of epigenetic control processes, stated in Section 2.3 are analyzed.

3.2 Classification of Elements of the Model According to Function as Information Carriers

In this section, we carry out a classification of the elements of the genetic apparatus according to their function in epigenetic control processes. The basic criterion for this classification is provided by the reduced dogma (see Crick [19]) of procaryotic genetics, stated in Section 2.1.

According to the dogma, three (3) classes of operational elements must be considered:

- Information Elements
- Transformation Elements
- Coordination Elements

Next, we discuss the most important characteristics of each of them.

The class of information elements is formed by all elements of the genetic apparatus involved in storage transcription and/or translation of genetic information. The most important operational characteristic of the elements in this class is that under interaction with other molecules or actions, they preserve their identity. For example, an operon does not change its identity under interaction with other elements; this is also true for MRNA's ribosomes and proteins.

The class of information elements is divided into three subclasses these are:

- Genetic Storage Subclass
- Transcriptor Subclass
- Protein Synthesizer Subclass

The genetic storage subclass is composed of the different operons in the system under study together with their addresses in the genetic locus of the system under study (see Section 2.1).

The basic feature of elements in the storage subclass is determined by the flow of information between these elements and the elements with

which they interact. As can be inferred from the operational diagrams of Figures 2.1.1 or 2.1.6, information always flows from an element of the genetic storage subclass to the elements that interact with it[1], and no information flows in the opposite direction. In synthesis, in our model, operons are assumed to be information invariant under interaction.

The transcriptor subclass is composed of two types of elements transcriptors and transcriptees. The transcriptors are elements representing RNA polimerase molecules[2]. An MRNA polymerase interacts with an element of the genetic subclass (if it is in stand-by intensity; see Figure 3.1.5). They trigger the transition of a transcriptee from a passive state to an active state so as to represent the information stored in the respective operon. In this interaction, also participates an element representing the respective element of the transcription feedback subclass which is described below.

We note that the transcriptor subclass is included in the model because of the fact that the most important control mechanism in an epigenetic process occurs at the transcriptional level. (see Lewin [11])

The active transcriptee represents the MRNA encoding the respec-

---

[1]This is an approximation. In actual procaryotic genetics there are mechanisms for Inverse transcription (see Baltimore and Spector [10]), but in this study we will assume that operons are not modifiable under interaction.

[2]Although there is evidence that at least two external factors (in addition to RNA polymerase) are required for transcription in our model we consider RNA-polymerase and all the required factors for transcription, as a single element. (see Watson [9] for a discussion of these external factors.)

tive operon. It carries among other things, (see Section 3.3) the number of different proteins encoded in the operon, and in interactive conjunction with an element of the clock-subclass discussed below, the time of synthesis[1] for each molecule.

The protein synthesizer subclass, as its name indicates, includes all the elements representing proteins[2] and elements involved in the synthesis of proteins; that is, ribosomes.

A protein is represented in our model by a single element the timing in the sequential synthesis of a protein (i.e., the incorporation of appropriate tRNA residue) that we call time of synthesis is simulated as the interaction of an element representing a ribosome with an element of the clock subclass which stores the time of synthesis of the protein under construction. The time of synthesis for each protein coded by the operon is stored in the corresponding transcriptee, as indicated above.

Now, we define the elements in the transformation class.

These elements represent the molecules upon which the epigenetic control processes act. They are the elements of the reaction systems introduced in Chapter 2.

From the operational point of view, the transformation elements are further classified into two subclasses:

---

[1] See Section 3.3.

[2] The proteins that are represented in the protein synthesizer subclass are those proteins whose synthesis is activated by a response to a transcriptional control action in the cell, i.e., we exclude those proteins whose synthesis is constitutive, i.e., independent on the conditions of the intra-cellular environment (see Watson [9]).

- Genetically Controlled Subclass

- Non-genetically Controlled Subclass

The genetically controlled subclass is formed by the elements which participate in reaction systems controlled by proteins non-constitutively synthesized; that is, proteins whose synthesis is a (controlled) response to specific conditions in the intra-cellular environment, e.g., the presence or absence of a particular metabolite.

The non-genetically controlled subclass is composed of those elements which participate in reaction systems regulated by constitutive proteins. These systems, in general, are indirectly controlled by the non-constitutive systems via local interaction with elements of the genetically regulated subclass.

In using our model for analyzing a particular epigenetic control process involving interaction[1] of a subset of the operon systems of the cell, all the transformation elements belonging to operon systems outside of the process are considered as non-genetically controlled elements. We remark that these elements affect the behavior of the process under study because they utilize common elements (resources) such as ribosomes and also possibly, the reaction systems of the processes interact with elements of the reaction systems of the other operons in the cell.

Finally, we describe the basic characteristics of the coordination class of elements. These elements do not represent molecules, but control or coordination actions in the genetic apparatus of the cell; for this reason, we refer to the elements of this class as virtual elements. (see Chapter 2).

[1]This interaction evolves according to the element organization discussed in Section 2.3 (Figures 2.3-1 and 2.3-6).

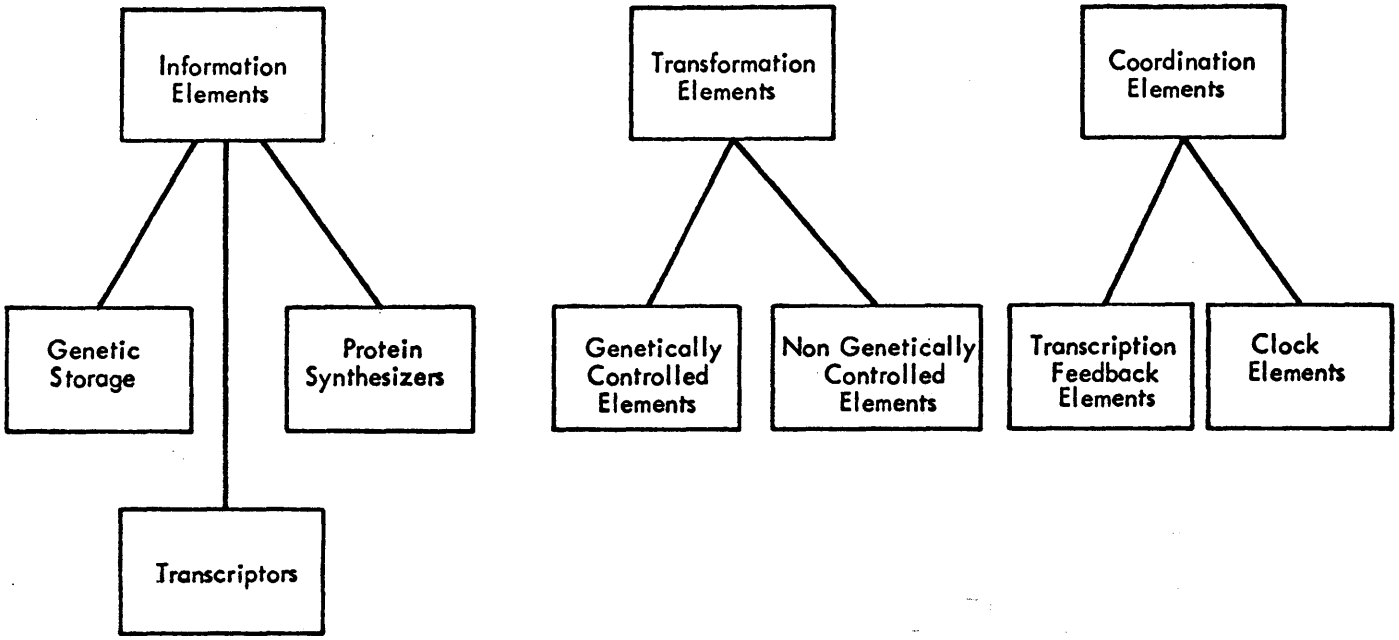
The class of coordination elements is divided into two subclasses; these are

- Transcription Feedback Subclass
- Clock Subclass

The transcription feedback subclass includes elements that carry out the control of the transcription of operons in epigenetic processes (see Section 2.1). These elements interact with elements of the transformation class, and with elements of the protein synthesizer subclass (i.e., regulator proteins) its denomination comes from the fact that in some operons, the activity of the operon is controlled by the products or (substrates) of its reaction system. As can be inferred from the example of Figure 2.1-6, this is not always the case, but since the behavior of transcription feedback elements has the effect of transmitting information from the transformation elements to the transcriptor elements, this denomination is justified.

The clock subclass, as its name indicates, is composed of elements whose function is to provide time-coordination in the dynamic interaction of elements in epigenetic processes, in particular, clock elements are used to represent asynchrony among subprocesses in a composite process (as discussed in Section 2.3), to simulate time of synthesis for protein elements and to establish time hierarchies of processes as will be explained in Chapter 4.

The classification of elements of the model according to the operational characteristics of the molecules or actions of the genetic apparatus they represent is summarized in the block diagram of Figure 1.



Block Diagram of the Operational Classification of the Elements in the Model

Figure 1

3.3 Incidence Diagram of Dynamic Interaction Among the Elements of the Classes

In this section we develop state diagrams for typical elements (types) of each of the subclasses established in the last section. The diagrams are obtained, for each type, from an analysis of the operational characteristics of the corresponding element in the genetic apparatus.

The state diagrams alluded above are the core of our model. They specify for each type the one-step (in time) structure and transition functions; these are given, in tabular form, for each type. In addition, an assignment of elements to a 3-dimensional information space is developed, and, on the basis of it, the input selector and neighborhood functions for each type are determined.

We start with the development and analysis of the type of the genetic storage subclass, a partical version of which was introduced in Section 3.1 for illustrative purposes.

Two major aspects are to be considered in the representation of the dynamics of genetic storage elements (i.e., operons)

-- Control of the release of information

-- Information Transcription

The state diagram corresponding to the control of information release, that is, the transcriptional control, was discussed as an example in Section 3.1 (see Figure 3.1-5 and accompanying discussion).

We now proceed to describe the state structure of the portion of the operon element involved in information transcription. Two questions must

be answered: What information is to be stored in the operon and how this information is transferred to the element representing the m-RNA polymerase. In order to answer these two questions, we consider simultaneously the representation of an ensemble composed of an element simulating the operon (only the information transcription part of it), an element simulating the M-RNA polymerase and the portion of the state structure of an element representing M-RNA, which is involved in transcription (the other part of the state structure of this element is involved in protein synthesis and will be described later on in this section).

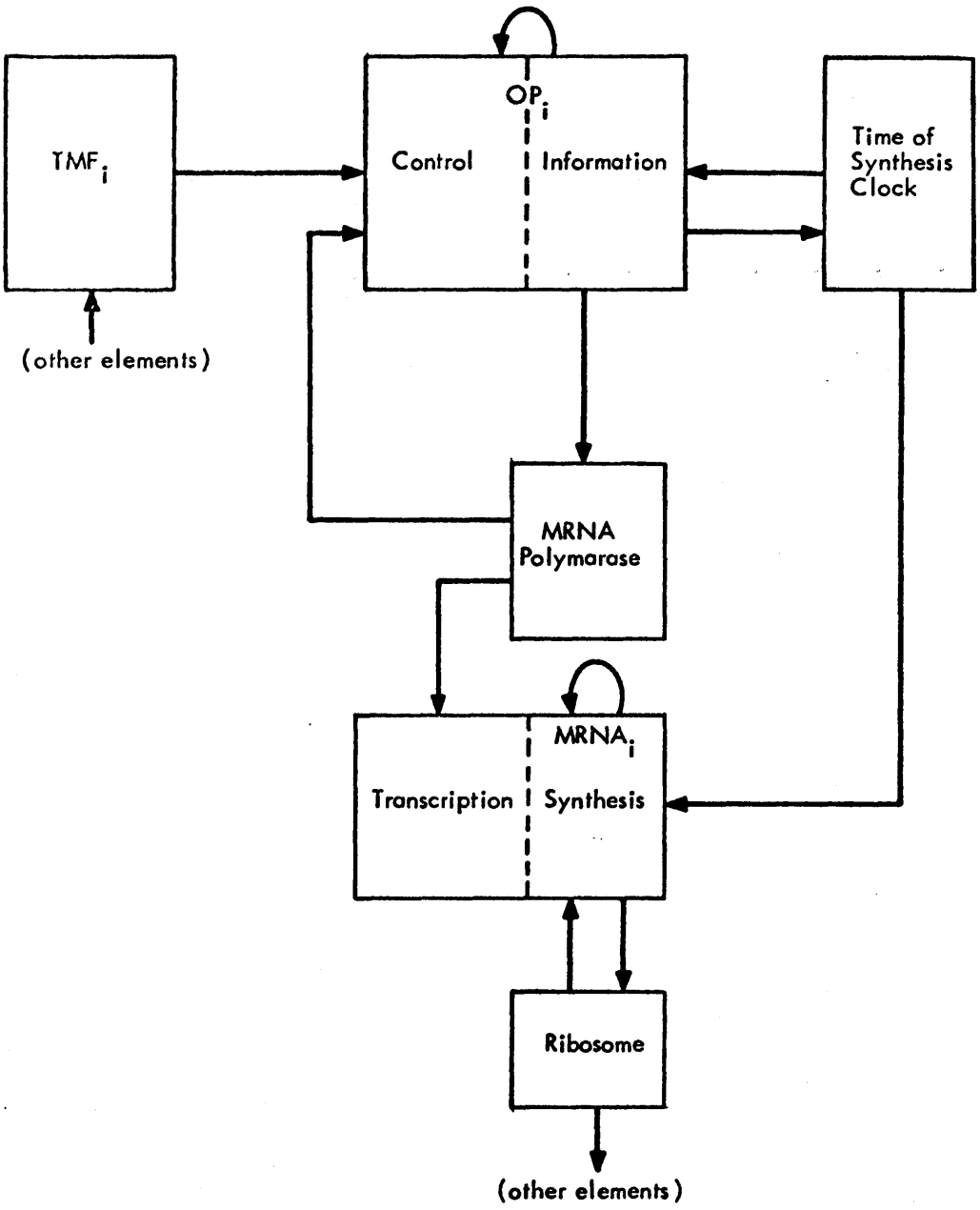
Figure 1 shows a scheme of the information flow among the elements of the ensemble defined above.

The information part of OP_i carries the following items: a) The number (n) of genes in OP_i, that is to say, the number of proteins coded by OP_i[1] and b) the length of each of these proteins.[2] We represent this information in the context of our model be assigning one structure for each gene and a corresponding intensity whose value (an integer) represents the length of the respective protein.

---

[1]Strictly speaking, a gene carries the code of a protein chain. A protein molecule is formed by one or more chains (see Lewin [11]). In our model, we assume $PR_{i,k}$, k=1,...,n to be "proteins" and simulate the formation of a true protein from the chains composing it (when the protein is composed of more than one chain) with chemically irreversible reaction steps of the types discussed in Chapter 2.

[2]In our study, we assume the length of a protein chain to be a positive integer which is functionally dependent of the actual number of amino-acid residues forming the chain. A criterion for determining this number is described in Section 3.5.



Information-Flow Between Elements in Transcription

Figure 1

In Figure 2, $T_{i,k}$, k=1,...,n are positive integers, each a function of the number of residues that compose the message for the corresponding protein, or rather, a function of the time that the M-RNA polymerase takes in transcribing the corresponding protein (we will see that each of these numbers will be taken in our model as the same time as that for the synthesis of the corresponding protein $PR_{i,k}$, k=1,...,n).[1]

Table 1 gives the part of the operon element structure transition function concerned with information storage. The structure C is the structure in the bottom of Figure 3.1-5. The numbers assigned[2] to the structures $G_{i,k}$, k=1,...,n are chosen according to the following criterion:

$$G_{i,k} \neq 0 \quad k=1,\ldots,n \quad \forall_i \text{ in the model}$$

$$G_{i,k} > G_{i,j} \quad \text{if } k > j \qquad (1)$$

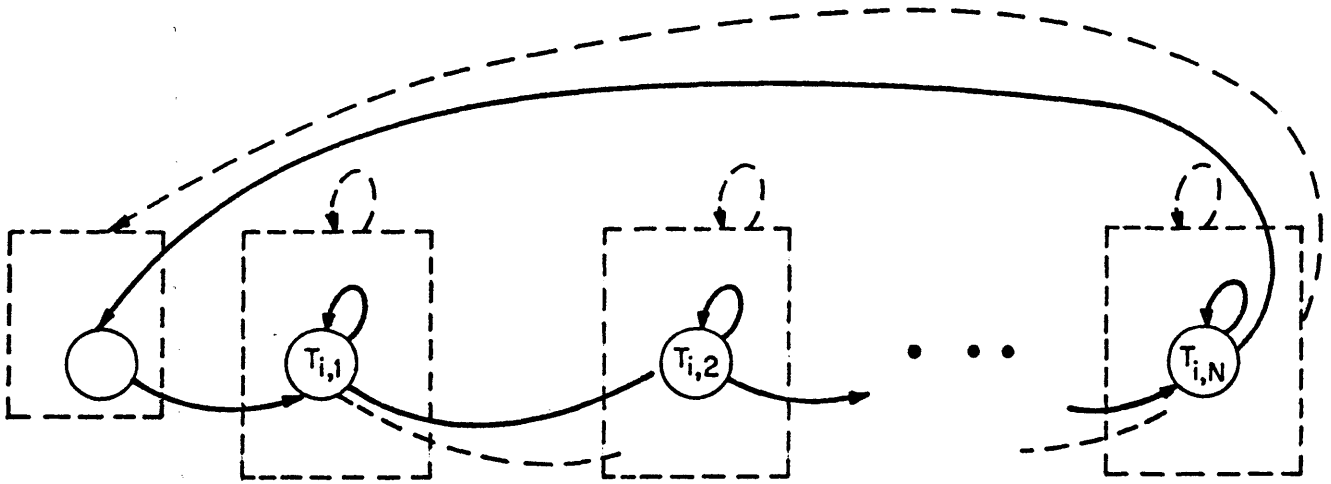
$$G_{i,k} \neq C \quad k=1,\ldots,n$$

The intensity transition functions for the type of Figure 2 are given in Table 2.

---

[1] In the procaryotic cell, the transcription and synthesis times for a given protein chain are not equal in general; however, with respect to the time-horizon of the cell, these two times are of the same order of magnitude (see Sampson [12] and Rosen [13]) with the synthesis time larger than the transcription time. In our model we assume these two times to be exactly equal. This assumption does not affect significantly our operational study of control mechanisms in the genetic apparatus because locally the actions of a protein (catalysis in reaction systems and regulation in TFM's) do not depend on wither of these two times. However, this assumption introduces some distortion in the structural identification problem (see Chapter 5).

[2] The criterion for assigning numbers to identify structures and intensities for each of the types is given as each type is introduced.



State representation of an operon element

Boxes represent structures, circles represent intensities, solid arrows represent intensity transitions, dashed arrows represent structure transitions

Figure 2

Table 1

Structure Transition Function for Operon Element

Present Structure (OP_i)	Present Intensity (OP_i)	Present Intensity (Clock)	Next Structure (OP_i)
G_i,k $k=1,\ldots n-1$	T_i,k	$T, T \neq T_i,k$	G_i,k
G_i,k	T_i,k	T_i,k	$G_i,k+1$
G_i,n	T_i,n	T_i,n	C
C	READ	0	$G_i,1$
C	S|by	0	C

Present Structure (OP_i)	Present Intensity (OP_i)	Present Intensity (TFM_i)	Next Structure OP_i
C	S|by	INA*	a,b,e, or d**
C	S|by	A*	C

*INA and A are intensities of the transcription feedback mechanism element. Its operational significance will be explained later in this section.

**a,b,e,d are the structures of the control part of the operon (see Figure 3.1-5 and companion discussion) these structures and their dynamics will be discussed after the discussion of the state structure of the TFM element which control their dynamics.

NOTE: For ease of readability we will herein write T_{i,k_i}, A_{i,k_i} etc... as T_i,k_i, A_i,k_i etc...

Table 2

Intensity Transition Functions For Operon
(information part)

Structure G_i,k $k=1,\ldots,n-1$

Present Intensity (OP_i)	Present Intensity (Clock)	Next Intensity (OP_i)
T_i,k	$T \neq (T_i,k)$	T_i,k
T_i,k	T_i,k	$T_i,k+1$

Structure G_i,n

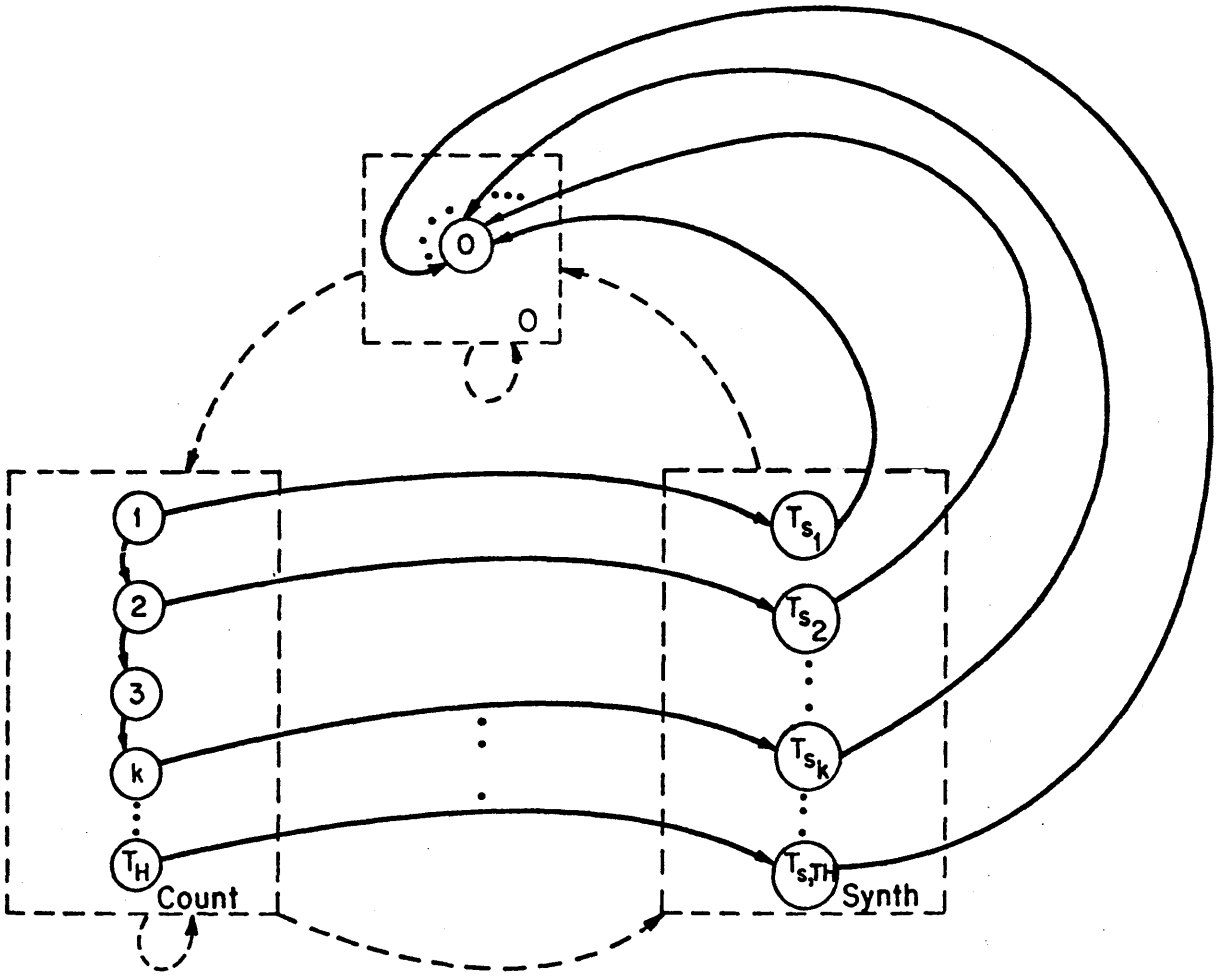
Present Intensity (OP_i)	Present Intensity (Clock)	Next Intensity (OP_i)
T_i,n	$T \neq (T_i,n)$	T_i,n
T_i,n	T_i,n	S|by

C (Control Part)*

Present Intensity (OP_i)	Present Intensity (TFM_i)	Present Intensity (Clock)	Next Intensity (OP_i)
S|by	INA	any	OFF**
S|by	A	0	ON**
ON	-	-	READ

*The control part will be discussed after introducing the state structure of the TFM element. We include here this table as an aid for our discussion of the information part.

**see Figure 3.4-5.



Time of Synthesis Clock (Virutal Element)

State Diagram

Figure 3

The structure transition C has two intensities S|by and READ. The state (C, S|by) of the operon element indicates that the operon is interacting with the corresponding TFM in active status (the state structure of TFM will be described later in this section) and waiting for the appropriate intensity of the nearest-neighbor[1] time-of-synthesis clock (the dormant intensity 0). When the intensity of the corresponding time-of-synthesis clock becomes 0, the operon suffers a state transition from (C, S|by) to (C, READ) and the transcription cycle proceeds as indicated above (see Tables 1 and 2).

Following the information diagram of Figure 1, we now pass to describe the type representing the time-of-synthesis clock. The state diagram of its type is shown in Figure 3 and its structure transition function and intensity transition functions are given in Tables 3 and 4, respectively.

The type representing the time-of-synthesis clock has 3 structures: count, synth, and the dormant structure (0) which was introduced earlier (Section 2.3) and whose properties are discussed in Section 4.1. When the clock is in the count structure, (so called because the clock, when in this structure is counting elapsed time steps in the reading of the operon by the m-RNA polymerase) at each step in time, it increases its intensity by one (i.e., if at time t the intensity is T, at time t+1 the

[1]See Figure 5 for relative allocation of operon elements and their time-of-synthesis clocks, to the informational space.

Table 3

Structure Transition Function for Clock Type

Present Structure	Present Intensity (Clock)	Present Intensity (OP_i)	Next Structure (Clock)
0	0	any $\neq$ S by	0
0	0	S by	count
count	$T \neq T_i,k$	T_i,k	count
count	$T = T_i,k$	T_i,k	synth
synth	any	$T_s,k = T_i,k$	0

for T every, $T \in \{1,\ldots,T_H\}$

Table 4

Intensity Transition Functions for Clock Type

Structure 0

Present Intensity (Clock)	Present Intensity (OP_i)	Next Intensity (Clock)
0	S|by	1
0	any $\neq$ S|by	0

Structure Count

Present Intensity (Clock)	Present Intensity (OP_i)	Next Intensity (Clock)
$T \neq T_H$*, $T \neq T_i,k$	T_i,k	T+1
T_i,k	T_i,k	$T_s,k = T_i,k$

Structure synth

Present Intensity (Clock)	Next Intensity (Clock)
T_s,k k=1,...,T_H	0

*T_H is the time horizon of the system.

intensity is T+1) until the intensity reaches the value $T_{i,k}$ which is the intensity assigned to the gene $G_{i,k}$ (see Figure 2) under transcription. This behavior is described by the intensity transition table of structure count given in Table 4.

When the clock in structure count, reaches an intensity equal to $T_{i,k}$ (after $T_{i,k}$ steps in time), a transition to the structure synth occurs. As indicated in Figure 3, for each intensity in count we have an intensity in synth. The intensity $T_{s,k}$ reached as indicated, is used by the M-RNA polymerase type to mark the end of transcription of the gene $G_{i,k}$ (of OP_i) just transcribed. This information is now registered in the element representing the $M\text{-}RNA_i$, which will be described in a moment. We note that the clock remains in the synth step on 1 time step and then returns to the dormant structure.

In the structure count, the number of intensities T_H is equal to the time horizon of the system under study, i.e.,

$$T_H = T^1 \tag{2}$$

where T^1 was defined in Section 2.3.

In synthesis, the clock element is a counter. Each counting step corresponding to a reading step of the corresponding operon. For each step in the counting procedure, we have provided the element with a corresponding intensity $\{T_{s,k}\}$ which serves as a signal for marking the completion of the transcription of a gene in the operon. The counting procedure starts always with the clock element in state (0,0) (See Fig-

ure 3) and is reinitialized after the completion of transcription of each gene in the operon.

The state diagram of the element representing the M-RNA polymerase is shown in Figure 4, the corresponding structure and intensity transition functions are given in Tables 5 and 6. The type has 4 structures labelled Noop, Act, Motion and UMotion. When the type is in the Noop structure and dormant intensity, the M-RNA polymerase is in an inactive status. This inactive status is changed by a transition to the Act structure (Act stands for active status) when the intensity of the operon (OP_i) under transcription is READ.

In the Act structure, the M-RNA polymerase type can be in one of two intensities: s and Br. In s, (which stands for synthesis), the transcription of OP_i is proceeding; this condition does not change until the intensity of OP_i becomes s by indicating that a transcription of one version of OP_i has been completed. When this is the case the intensity changes to Br (Br stands for Break). The functional objective of the intensity break is to indicate to the $M\text{-}RNA_i$ type that the transcription of one copy of a gene of OP_i has been completed so that the last gene of OP_i to the $M\text{-}RNA_i$ type and the M-RNA polymerase type must separate from each other (i.e., move apart by one unit in the informational space as explained below). This separation involves two steps:

-- The type changes its structure to Motion (see Figure 4) and to intensity R (which stands for Reset)

Table 5

Structure Transition Function for M-RNA Polymerase Type

Present Structure (M-RNA Pol)	Present Intensity (M-RNA Pol)	Present Intensity (OP_i) (Neighbor 2)	Next Structure (M-RNA Pol)
Noop	0	set	set[1]
Noop	0	S| by	Noop
Noop	0	READ	Act
Act	S	T_i,k k=1,...,n	Act
Act	S	S| by	Act
Act	Br	S| by	Motion
Motion	R	any	Motion
Motion	0	any	Noop

*U Motion

[1]When M-RNA Pol is in downwards position

Table 6

Intensity Transition Functions for M-RNA Pol. Type

Noop

Present Intensity (M-RNA Pol)	Present Intensity (OP_i) (Neighbor 2)	Next Intensity (M-RNA Pol)
0	S|by	0
0	READ	S
set	-	0

Act

Present Intensity (M-RNA Pol)	Present Intensity (OP_i) (Neighbor 2)	Next Intensity (M-RNA Pol)
S	T_i,k k=1,...,n	S
S	READ	Br
S	S|by	0
Br	S|by	R

Motion

Present Intensity (M-RNA Pol)	Next Intensity (M-RNA Pol)
R	0

Table 7

Structure Transition Function of M-RNA pol. for Upward Motion

Present Structure (M-RNA Pol)	Present Intensity[1]				Next Structure (M-RNA Pol)
	Neighbor 0	Neighbor 1	Neighbor 2	Neighbor 4	
Noop	0	0	0	0	Noop
Noop	0	De[2]	0	0	UMotion
UMotion	set	De	0	0	Noop
UMotion	--	--	--	set[3]	Noop

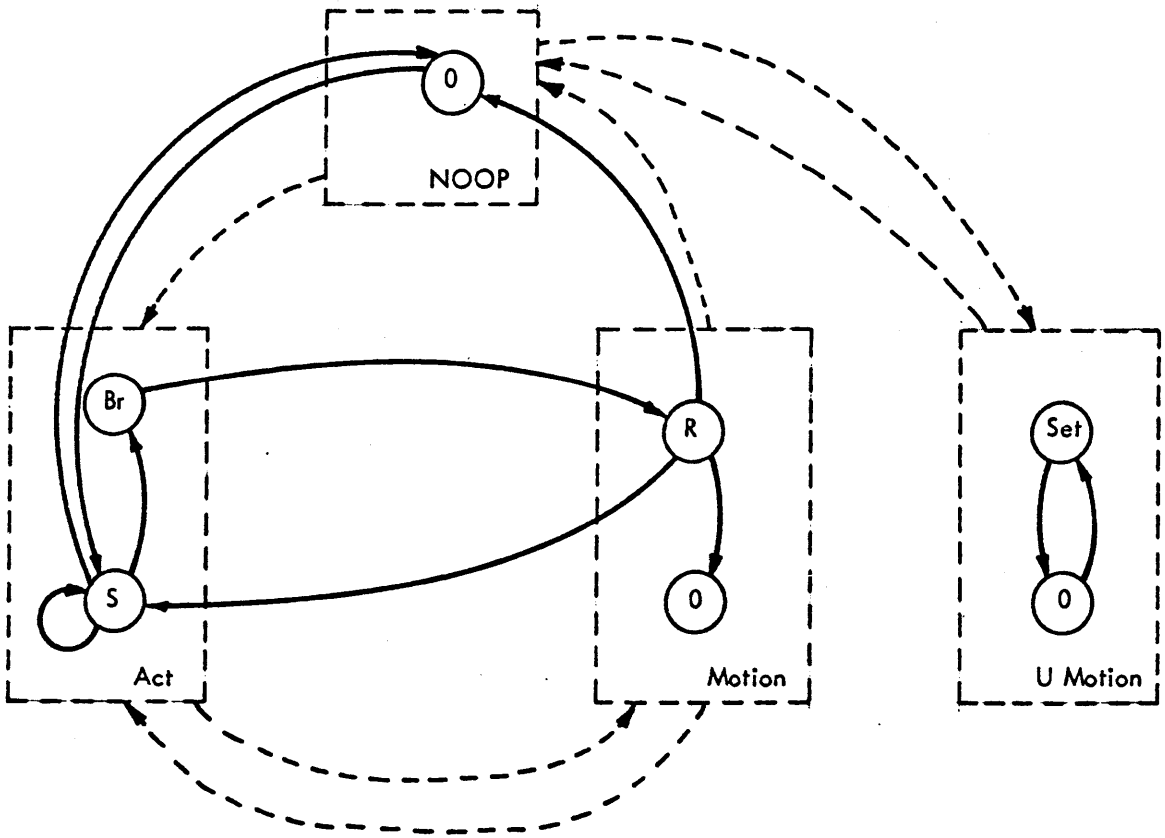
Intensity Transition Function for Upward Motion (UMotion)

Present Intensity (M-RNA Pol)	Present Intensity			Next Intensity (M-RNA Pol)
	Neighbor 1	Neighbor 2	Neighbor 4	
0	De	0	0	set
set	De	0	0	0
0	--	--	set	0

[1] See Figure 5a for neighbor-numbering system.

[2] De is an intensity of the type M-RNA$_i$ and also the intensity of a virtual element whose only state is (De$_i$, De) (see Figure 5).

[3] This is the intensity of the M-RNA pol. element left behind when the element does its downard-motion.



State Diagram of the MRNA Polymerase Type

Figure 4

-- When in structure Motion and type R, the M-RNA polymerase changes its state (autonomously; see Tables 5 and 6) to the state (Noop, 0), after a motion in the z_2-direction in the information space. In order to understand the mechanics of this step we need to specify the relative positions of the elements involved in transcriptional control of our model. These positions are shown in a two-dimensional projection of the (3-dimensional) information space illustrated in Figure 5.

The operons of the system under study are allocated in the z_1-axis ($z_2 \equiv 0$) in the locations $z_1 = 1$, for operon 1 (OP_1), $z_1 = 3$ for operon 2, $z_2 = 5$ for operon 3 and so on. The corresponding clock elements are allocated in coordinates (2,0), (4,0), (6,0), etc. The corresponding M-RNA elements, $M\text{-}RNA_1$, $M\text{-}RNA_2$, $M\text{-}RNA_3$ are allocated in the coordinates (2,-1), (4,-1), (6,-1) and so on, and the other elements of Figure 1 are allocated so as to preserve the nearest neighboring condition. With this preliminary description, now we are ready to explain the reset motion. Once the M-RNA polymerase has completed the transcription of the $M\text{-}RNA_i$, it moves downwards in the z_2-direction so as to disattach[1] itself from

[1]This motion eliminates the nearest neighboring condition that existed between $M\text{-}RNA_i$, OP_i, and $M\text{-}RNA_{pol}$. The motion actually is simulated by a transfer of the state of the element to an element which is in dormant state and allocated directly below the M-RNA polymerase element in the z_2-direction. The former becomes the element representing the M-RNA polymerase while the latter becomes an element in dormant state.

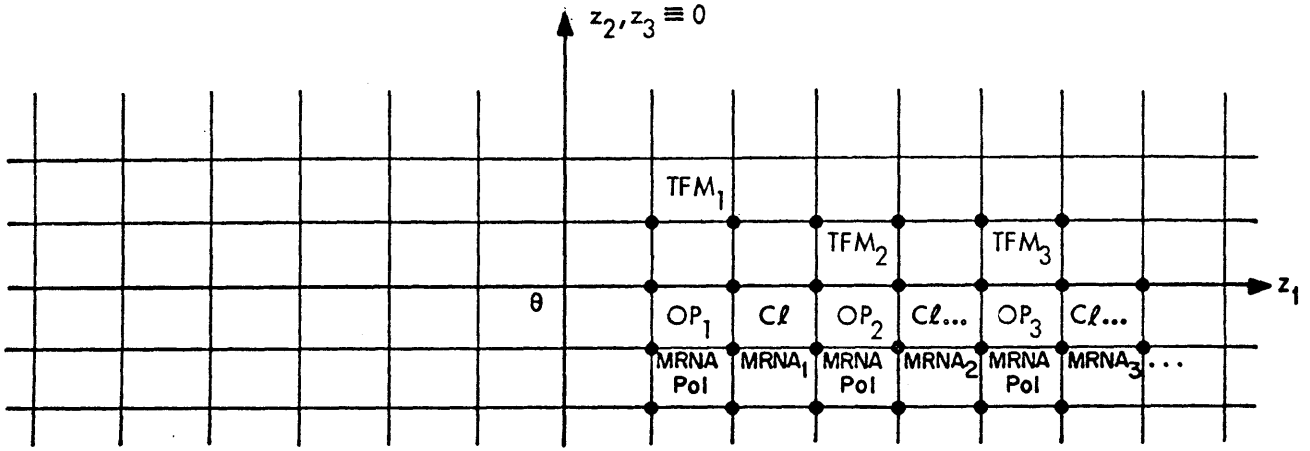
both the operon under reading and the transcribed M-RNA. Notice that that M-RNA-polymerase returns to the (Noop,0) state and thus it can be reused again. In order for that to happen, it must do an upwards motion[1] (in the z_2-direction) the structure transition function of Table 5 is then completed with this motion, which is given in Table 7.

The upwards motion is triggered by M-RNA$_i$ when the M-RNA polymerase is in intensity De. The functional significance of this will become clear once we describe the state structure of the M-RNA type.

Before we pass to the discussion of the state transition structure of the M-RNA type we stress an important point about element motion in our model. Elements in our models are assigned to fixed locations in the information space when motion of an element, such as the one discussed above for the M-RNA polymerase type, is required we simulate it by a transfer of its present memory contents (i.e., its present state) to the appropriate nearest-neighbor element (see footnote) this implies that the state of that element is changed to the one of the element whose motion is being simulated. This requires as set of conventions for state transfers; these conventions are given in the form of a virtual element towards the end of this section.

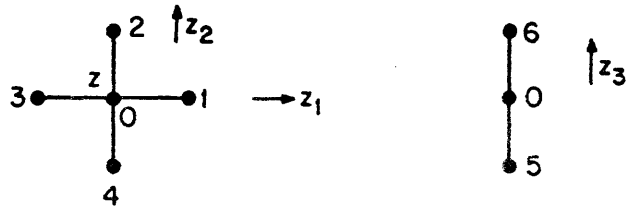
---

[1]The upwards motion is the symmetric counterpart of the downwards motion; the M-RNA polymerase is allocated in a position with z_2-ordinate equal to -2 and in state (UMOTION, SET) and suffers an autonomous transition to (Noop,0) while moving upwards one unit in the z_2-direction.



(a) Allocation of elements involved in transcriptional control in the z_1-z_2 plane ($z_3 \equiv 0$) of the informational space

Figure 5



(b) Neighbor numbering of an element allocated in a 3-dimensional information space

Figure 5a

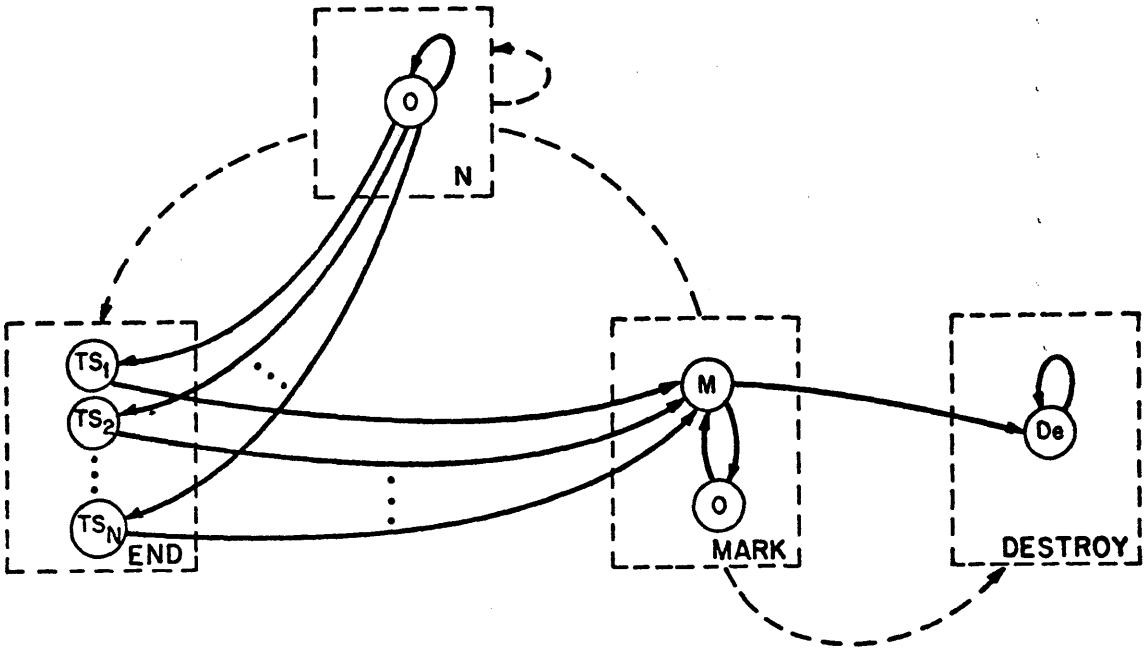
Now we consider the state transition structure of the M-RNA type. A state diagram of this type is shown in Figure 6. The corresponding structure and intensity transition functions are given in Tables 8 and 9, respectively.

The M-RNA type which is allocated in the informational space as indicated in Figure 5 is not a nearest neighbor of the corresponding operon. All the information corresponding to the proteins coded by the operon, is transferred to the M-RNA element by the M-RNA polymerase element which is a nearest neighbor of both the operon and the M-RNA elements.

The M-RNA type has 4 structures labelled N, END, MARK, and DESTROY. In the N structure the M-RNA type is in neutral condition. The type is either inactive or following the transcription from an operon type (say OP_i) to the corresponding clock, as explained before. The intensity during information transfer is the dormant intensity. When the clock detects the completion of transcription of a gene (say $(G_{i,k})$, the intensity of the clock becomes $T_{s,k}$, for some k, and this triggers a structure transition in the M-RNA type from N to END, and a corresponding intensity transition from 0 to $T_{s,k}$, as shown in Tables 8 and 9.

We note that the only feature of our M-RNA elements for distinguishing between different proteins consists of the times of synthesis $\{T_i,k_i\}$.

When in structure END, and intensity $T_{s,k}$, k=1,...,m, any intensity of the clock in structure count, will cause the type to change it structure to MARK and intensity M. The operational object of this state is



State Diagram of the MRNA Type

Figure 6

Table 8

Structure Transition Function for M-RNA Type

Present Structure (M-RNA)	Present Intensity (M-RNA)	Present Intensity (Clock)	Present Intensity (M-RNA Pol)	Present Intensity (Ribosome)	Next Structure (M-RNA)
N	0	$T, T=1,\ldots T_H$	0	any	N
N	0	$T_{sk}\ k=1,\ldots M$	0	any	END
MARK	M	any	Br	cont	MARK
MARK	M	any	S	cont	N
MARK	M	any	Br	Decont	Destroy
Mark	0	any	0	cont	N
Destroy	De	any	any	any	Destroy
END	$T_{sk}\ k=1,\ldots T_H$	$T, T=1,\ldots T_H$	any	any	MARK

Table 9

Intensity Transition Functions for M-RNA Type

N

Present Intensity (M-RNA)	Present Intensity (Clock)	Present Intensity (M-RNA Pol)	Next Intensity (M-RNA)
0	0	0	0
0	T, T=1,...,T_H	0	0
0	T_{sk} k=1,...,M	0	T_{sk}

MARK

Present Intensity (M-RNA)	Present Intensity (M-RNA Pol)	Present Intensity (Ribosome)	Next Intensity (M-RNA)
M	Br	cont	0
0	set	cont	0
M	any	Decont	De
M	R	cont	M

Destroy

Present Intensity (M-RNA)	Next Intensity (M-RNA)
De	De

END

Present Intensity (M-RNA)	Present Intensity (Clock)	Next Intensity (M-RNA)
T_{sk} k=1,...,M	any	M

to inform the ribosome element that a protein has been transcribed, and the length of the protein under synthesis[1].

When in structure MARK, and intensity M, the M-RNA has completed the transcription of a protein. If the operon under transcript has more genes to be transcribed, the M-RNA polymerase is in intensity Br and if the ribosome is in the intensity count[2], the transcription of the next gene of the operon starts. This is simulated by a state transition to the state (N,0) starting a new iteration of the cycle described in the last paragraphs.

When in structure MARK, and intensity 0, if the M-RNA polymerase is in set intensity, the M-RNA attains the state (N,0); that is, the information about the last gene transcribed is "erased"[3] and the M-RNA type is ready for the transcription of the next gene. On the other hand, if the M-RNA polymerase is in the R intensity, the M-RNA moves downward (along the z_2-coordinate) in the informational space disattaching itself

---

[1]We note that our model the transcription process and the corresponding synthesis process of a given protein operate simultaneously, a condition that is known to hold in the cell. (see Watson [ ]). That is, the portion of M-RNA transcribed at a given point can drive a ribosome for the synthesis of the corresponding chain.

[2]The operational significance of the intensity count of the ribosome type is explained later.

[3]This operation has no corresponding behavior in the genetic apparatus in which the operon is transcribed as a polycistronic M-RNA. However, from the operational point of view, the important aspect is the sequential transcription of the genes of the operon and this is preserved in the state transition convention described above.

from its clock (see Figure 5). Finally, if the M-RNA type is in the MARK structure and M intensity, and the ribosome is in the Decont[1] intensity, the next state of the type is (Destroy, De)[2]. This state indicates that the M-RNA type has reached as irreversible status which prevents it from feeding (via its intensity) any further information to the ribosome type. This condition of the type simulates the destruction (by external agents or by degradation; see Lewin [11]) of a polycistronic M-RNA molecule after it has served as the template for the synthesis of a number of proteins (see Section 2.3). Thus is the operon (OP_i) under synthesis is no longer fed by its corresponding transcription feedback mechanism (TFM_i), no further synthesis of its proteins occurs. If the TFM_i is still active[3] the transcription cycle described so far starts again.

Now, we will describe the ribosome type. In order to do this we first provide a diagram of information flow for the ensemble of elements that interact with the ribosome. This diagram is shown in Figure 7.

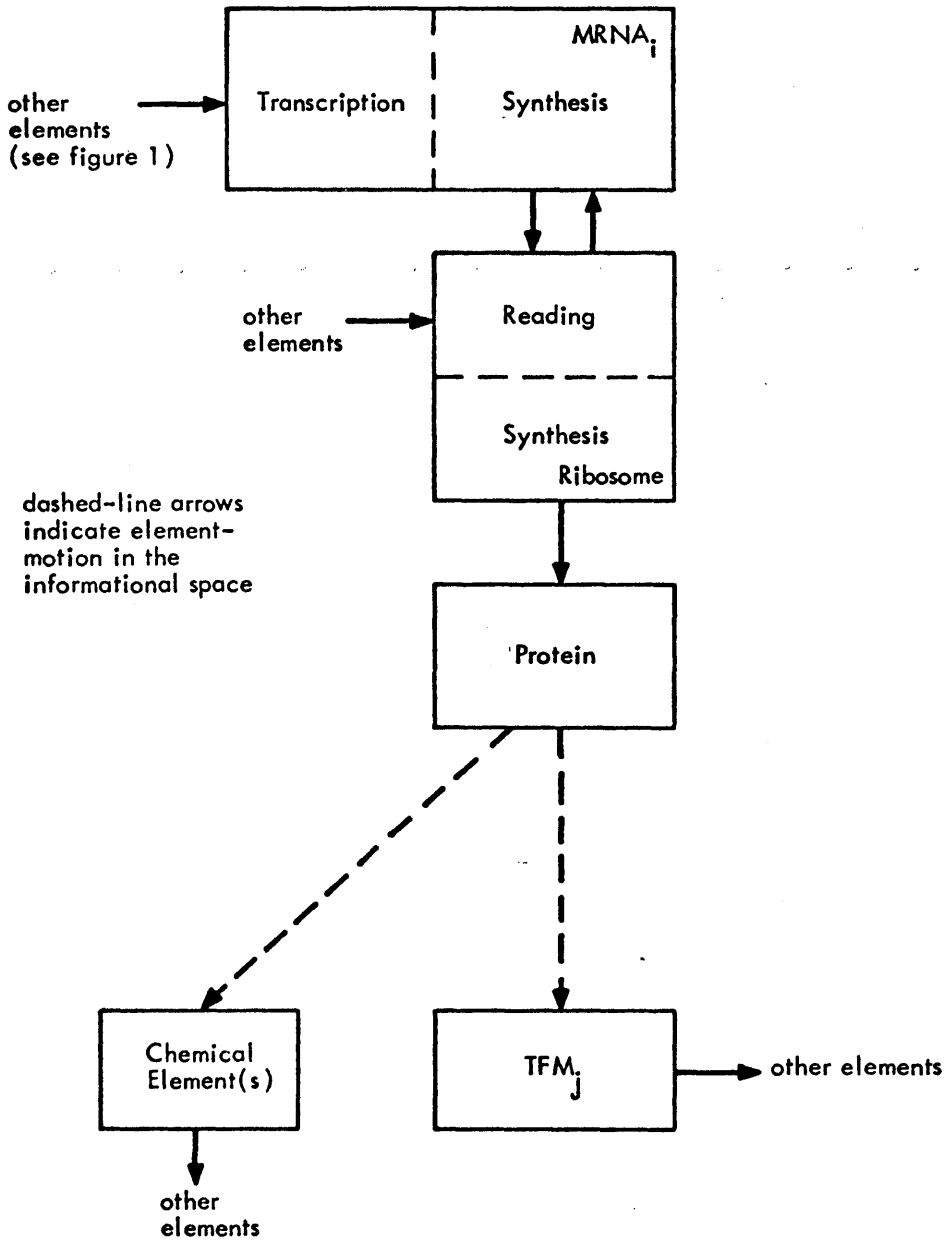
In Figure 8, the state diagram for the ribosome type is given. The corresponding structure and intensity transition functions are given in Tables 10 and 11, respectively. The allocation of ribosome elements to

---

[1]Decont is an intensity of the ribosome type which will be described below.

[2]The operational significance of intensity De was discussed before in connection with the $M\text{-}RNA_{pol}$.

[3]As most other elements described so far, the TFM_i has an inactive state all the other states of this type are referred to as active.



Information-flow Between Elements in Protein Synthesis

Figure 7

Table 10

Structure Transition Function of the Ribosome Type

Present Structure (Ribosome)	Present Intensity (Ribosome)	Present Intensity ($M\text{-}RNA_i$)	Next Structure (Ribosome)
INA_ℓ ℓ=1,...,p	ℓ ℓ=1,...,p	0	ℓ, ℓ=1,...,p
ℓ ℓ=1,...,p	cont	0	ℓ, ℓ=1,...,p
ℓ ℓ=1,...,p	cont	T_{sk} k=1,...,M	ℓ, ℓ=1,...,p
ℓ ℓ=1,...,p	cont	Br	ℓ, ℓ=1,...,p
ℓ ℓ=1,...,p-1	Br	any	$INA_{\ell-1}$ ℓ=1,...,p-1
p	Br	0	TRANS
TRANS	Decont	De	TRANS
TRANS	S1	0	INA_1
TRANS	R1	any	TRANS

[1]The domain of structure TRANS has 5 arguments, i.e., TRANS Q^5 Q where Q is the set of intensities; however the structure transition associated with TRANS depends only on 2 intensities, as indicated in the table.

Table 11

Intensity Transition Functions of the Ribosome Type

INA_ℓ $\ell=1,\ldots,p$

Present Intensity (Ribosome)	Present Intensity (M-RNA)	Next Intensity (Ribosome)
ℓ	any $\neq 0$	ℓ
ℓ	0	cont

ℓ, $\ell=1,\ldots,p-1$

Present Intensity (Ribosome)	Present Intensity ($M\text{-}RNA_i$)	Next Intensity (Ribosome)
cont	0	cont
cont	T_{sk}, some $k=1,\ldots,M$	Br
Br	M	$\ell+1$

P

Present Intensity (Ribosome)	Present Intensity ($M\text{-}RNA_i$)	Next Intensity (Ribosome)
cont	0	cont
cont	T_{sk}, some $k=1,\ldots,M$	Br
Br	any	Decont

Table 11 (contd.)

TRANS

Present Intensity (Ribosome)	Present Intensity				Next Intensity (Ribosome)
	Neighbor 5[1]	Neighbor 1	Neighbor 3	Neighbor 4	
Decont	De	Decont	cont	0	R_1
Decont	De	cont	Decont	0	S_1
Decont	De	Decont	Decont	0	R_1
Decont	De	R_1	Decont	0	R_1
Decont	De	R_1	S_1	0	Decont
Decont	De	R_1	R_1	0	R_1
Decont	De	S_1	S_1	0	S_1
Decont	Br	S_1	0	0	S_1
Decont	Br	R_1	0	0	S_1
R_1	De	any$\neq R_1, S_1$	any$\neq R_1, S_1$	STOP	S_1
S_1	0	0	0	0	1
R_1	0	0	0	0	S_1
R_1	Br	any	any	any$\neq 0$	S_1
R_1	0	R_1	R_1	0	S_1
R_1	0,or T_{sk} k=1,...,M	Decont	S_1	any	S_1
S_1	Br	S_1	S_1	0	1
S_1	De	R_1	R_1	any	R_1
S_1	0	cont	S_1	0	S_1
S_1	De	any$\neq S_1, R_1$	any$\neq R_1, S_1$	STOP	R_1

[1] See Figure 5a for neighbor-numbering convention; and Figure 9 for relative position of ribosome in informational space.

the informational space is shown in Figure 9. (Note that all the activity here is in the z_1-z_3 plane as opposed to the z_1-z_2 plane used in Figure 5.)

The ribosome element has 3 types of structures: INA_i, i=1,...,p, ℓ, ℓ=1,...,p and TRANS, where p 0 is the number of proteins a ribosome can synthesize from a given M-RNA encoding before breaking.[1] The structures of the first type present the ribosome in an inactive status and ready for starting the synthesis of a protein. The structures of the second type indicate that the ribosome is engaged in the synthesis of a protein coded by the M-RNA element interacting with it. The structure

---

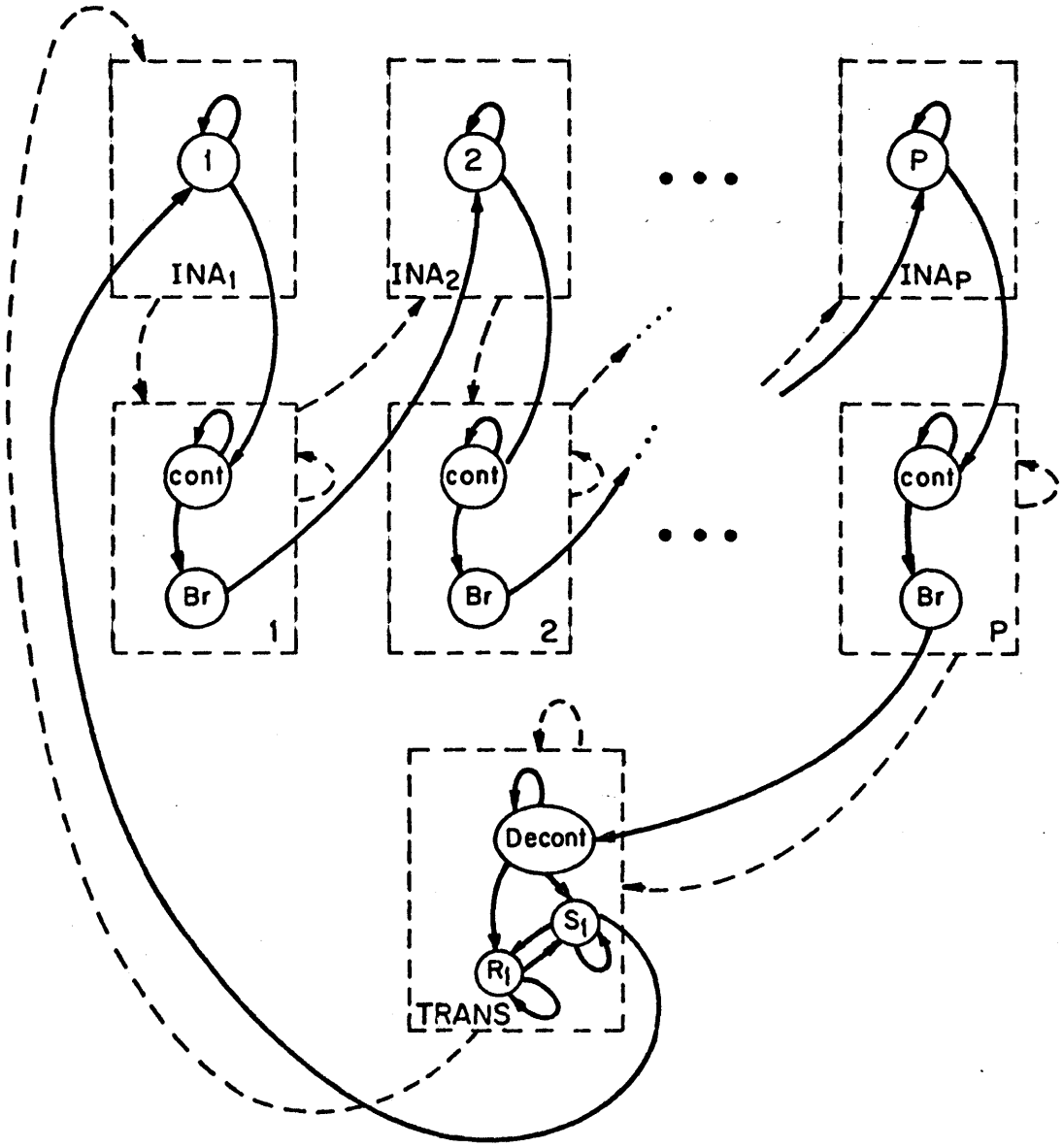
[1]It is well known that the ribosome in the cell is composed of two subunits (see Watson [9]) called the large subunit and small subunit; the former carries the synthesis and the latter reads the nucleotides of the M-RNA under synthesis. The two subunits are held together by a protein called the "sticky point". In the cell, (see Lewin [11]) there is usually a pool of ribosome subunits (presumably synthesized constitutively) which are assembled together when the "sticky points" are synthesized by a (presumably) regulated operon. A ribosome involved in synthesis suffers from irreversible sequential "fatigue" (Katchalsky [20]), that is, as more proteins are synthesized the likelihood of its subunits separating from each other increases. In our model of the ribosome, we simulate this fact by allowing each ribosome to synthesize p proteins before Breaking. The number p is also the max. number of proteins from a given polycistronic M-RNA that we allow to be synthesized before the M-RNA is destroyed. It should be noted that in our model at any given time only one ribosome is interacting with the M-RNA element in contract to what actually occurs in the cell, where many ribosomes are simultaneously engaged in the synthesis of proteins with a single M-RNA molecule as the Information carrier. However we simulate this fact by having a battery of ribosomes moving over the RNA, so that as soon as one version of M-RNA has been translated, another ribosome element is interacting with it. We will illustrate in Chapter 5 with a Monte-Carlo run example that, with this operational arrangement, the ratio of #Proteins/# respective M-RNA is about the same as that observed in the sel(20 $\neq$ 50).

TRANS indicate that the ribosome is moving along a path parallel to the z_1-axis ($z_2 \equiv -1$, $z_3 \equiv 1$), or, that the element is not available for protein synthesis. We now proceed to explain the operational role of these 3 types of structures and their corresponding intensities.

The structure INA_i, i=1,...,p indicates, in addition to the readiness of the ribosome type, the number of proteins synthesized by the ribosome since it was activated. When a ribosome is in structure INA_i, i=1,...,p $2 \leq i \leq p$, it means that is has synthesized i-1 proteins in the immediate past and is ready for the synthesis of the ith as soon as an RNA element interacts with it. Each structure INA_i has a single intensity (i).

When the ribosome is in state (INA_i, i) i=1,...,p and the M-RNA interacting with it is in intensity 0 a transition to state (i, cont) occurs and the synthesis of the corresponding protein proceeds.

In each of the structures labelled i, i=1,...,p the ribosome can attain one of two intensities: cont and Br (see Figure 8). When in state (i, cont), $1 \leq i \leq p$ the ribosome element is engaged in the synthesis of a protein element. This state remains unchanged during the synthesis step (which coincide with reading steps from the corresponding $M\text{-}RNA_k$). A state transition occurs when the intensity of the corresponding M-RNA element reaches intensity Br (indicating the completion of the code of a protein); the new state becomes (i, Br) which indicates completion of synthesis of the corresponding protein element. We note that the synthesis procedure discussed above proceeds sequentially until M-RNA element reaches



State Diagram of Ribosome Type

Figure 8

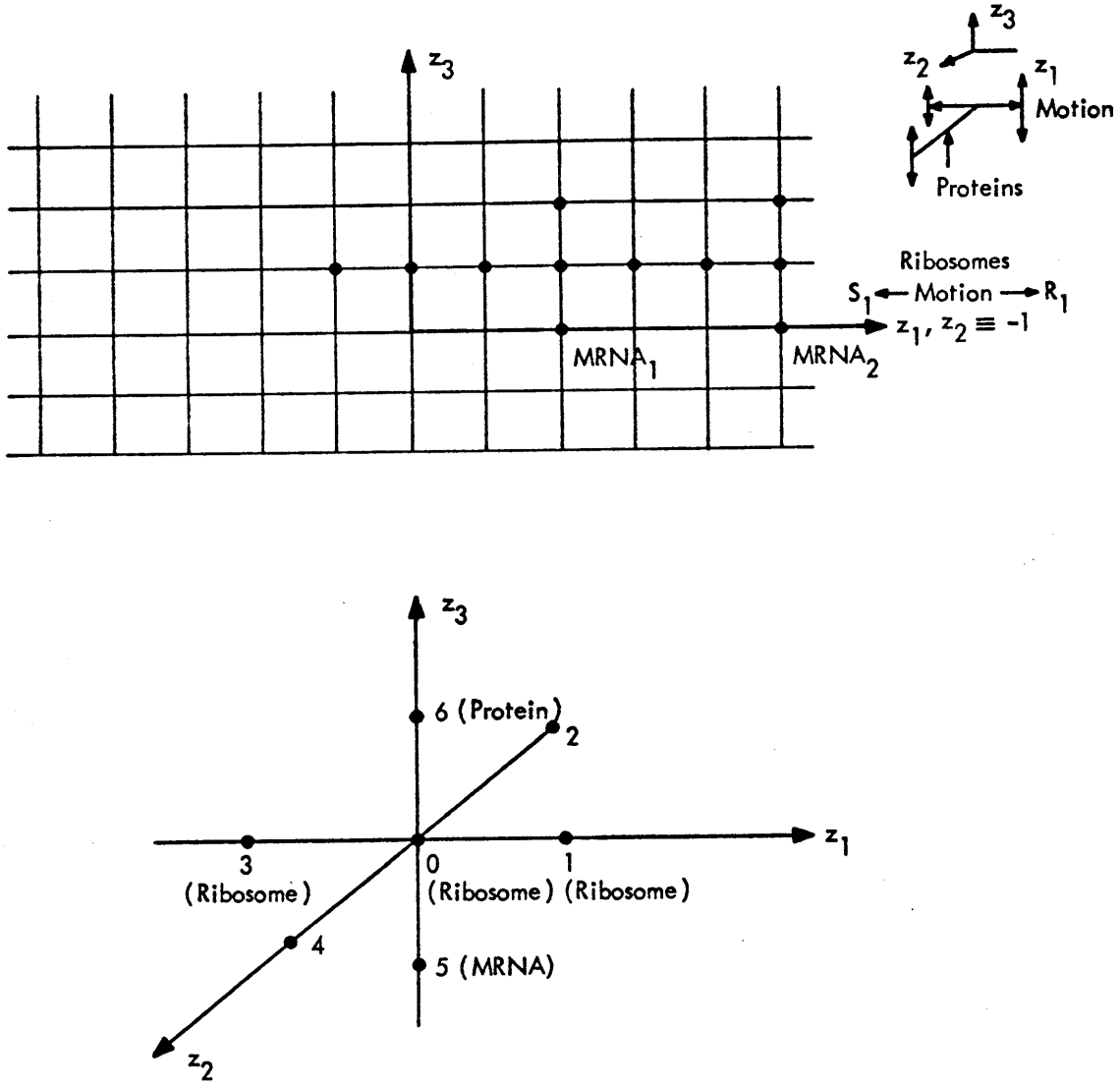
intensity M, then the protein element "knows" its synthesis has ended.

When in state (i, Br) $1 \leq i \leq p-1$, the ribosome element suffers an autonomous state transition to state (INA_{i+1}, i+1) and the ribosome element is ready for the synthesis of a new protein. When in state (p, Br) the ribosome has already synthesized p proteins in a role and, as discussed above, is unable to synthesize any more problems (breaking). In our model, this condition is represented by a transition to state (TRANS, Decont) as indicated in Figure 8.

The structure TRANS (whose mnemonics stand for transport) has 3 intensities: Decont, S_1 and R_1.

When in structure TRANS, the ribosome element is unable to synthesize proteins and is involved in a translation motion in the informational space as indicated in Figure 8. In intensity R_1 the translation motion is carried one space unit at each time step in the positive direction of the z_1-axis. This motion continues until one of the following conditions occurs:

a) An element representing an M-RNA in a nearest neighboring location of the point in which the ribosome element is at present time, is in 0 or $T_{s,k}$, k=1,...,M intensities meaning that the M-RNA is under transcription and a ribosome has not been made available for synthesis (the corresponding ribosome element directly above the M-RNA is in Decont intensity; see Fig. 9). In this situation a state transition of the ribosome-element occurs (from (TRANS, R_1) to (TRANS, S_1)) which positions the ribosome element in the appropriate nearest neighboring point of the M-RNA element (directly above



Allocation of Ribosomes in Informational Space and Their Relative Position w.r.t. Other Elements

Figure 9

it in the z_3-coordinate direction). When the ribosome is in state (TRANS, S_1) if the M-RNA element directly above is ready for synthesis (i.e., in intensity $\{T_{s,k}\}$) the ribosome element expreiences a state transition to state (FNA, 1) and synthesis starts as explained above This transition does not involve any motion.

b) A ribosome element in a nearest neighboring point (see Figure 9) is in intensity S_1[1] then the ribosome element changes its state to (TRANS, S_1).

c) An element in a nearest neighboring point of the ribosome element is in the intensity STOP. In this case, the ribosome suffers a state transition from (TRANS, R_1) to (TRANS, S_1). The intensity STOP belongs to an element allocated at a point $(z^1, 1\ \ 1)$[2] in the informational space; this element has no biological significance, its objective is to serve as a <u>barrier</u> (or boundary) for the motion of ribosome elements in the positive direction along the z_1-axis.

When in structure TRANS and intensity S_1 and when the neighboring M-RNA is not ready for synthesis, the ribosome element is in translation motion in the negative direction of the z_1-axis, one space unit at each time step. This motion continues until one of the following conditions occurs.

---

[1] As we shall see the intensity S_1 indicates translation in the negative direction of the z_1-axis.

[2] $z^1 \geq 3$

a') Condition a') is the symmetric of condition a) above (i.e., the ribosome is moving in the opposite direction of the z_1-axis). The corresponding state transition is from (TRANS, S_1) to (TRANS, R_1).

b') Condition b') is a symmetric version of condition b). The corresponding state transition if from (TRANS, S_1) to (TRANS, R_1).

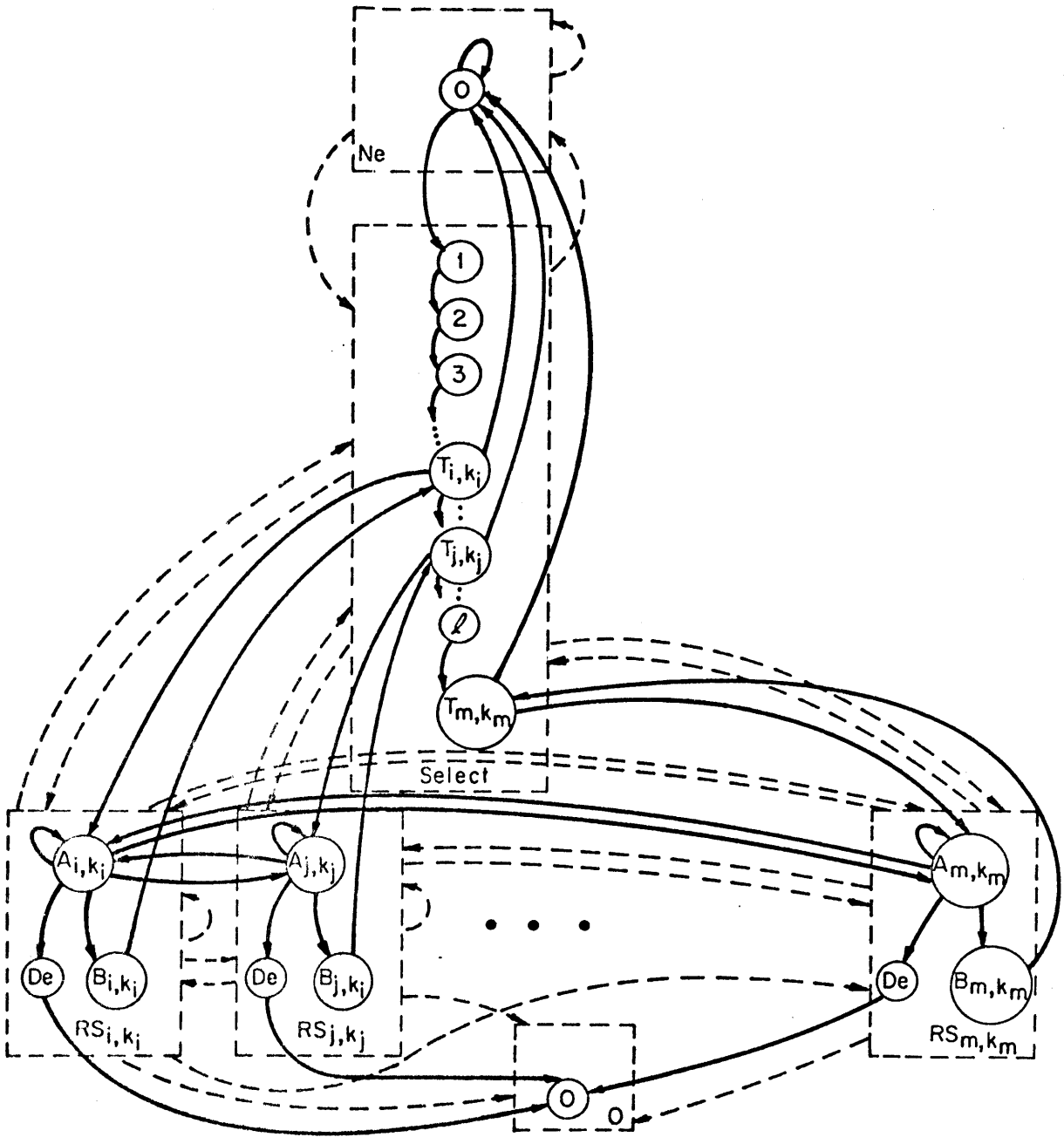
c') Condition c') is a symmetric version of condition c). The element STOP (i.e., a virtual element whose only state is (STOP, STOP) is allocated at some point (z^1, 1, 1) of the informational space with $z^1 \leq 3$.

In synthesis, the ribosome elements bounce back and forth along the z_1-axis when not engaged in synthesis. The motion is stopped when (1) a ribosome interacts with an M-RNA element which is in state (INA, 1), that is, ready for synthesis, or, (2) it finds an element STOP which causes the motion to reverse direction.

We note that the ribosomes move in the z_1 direction back and forward between the two stop elements.

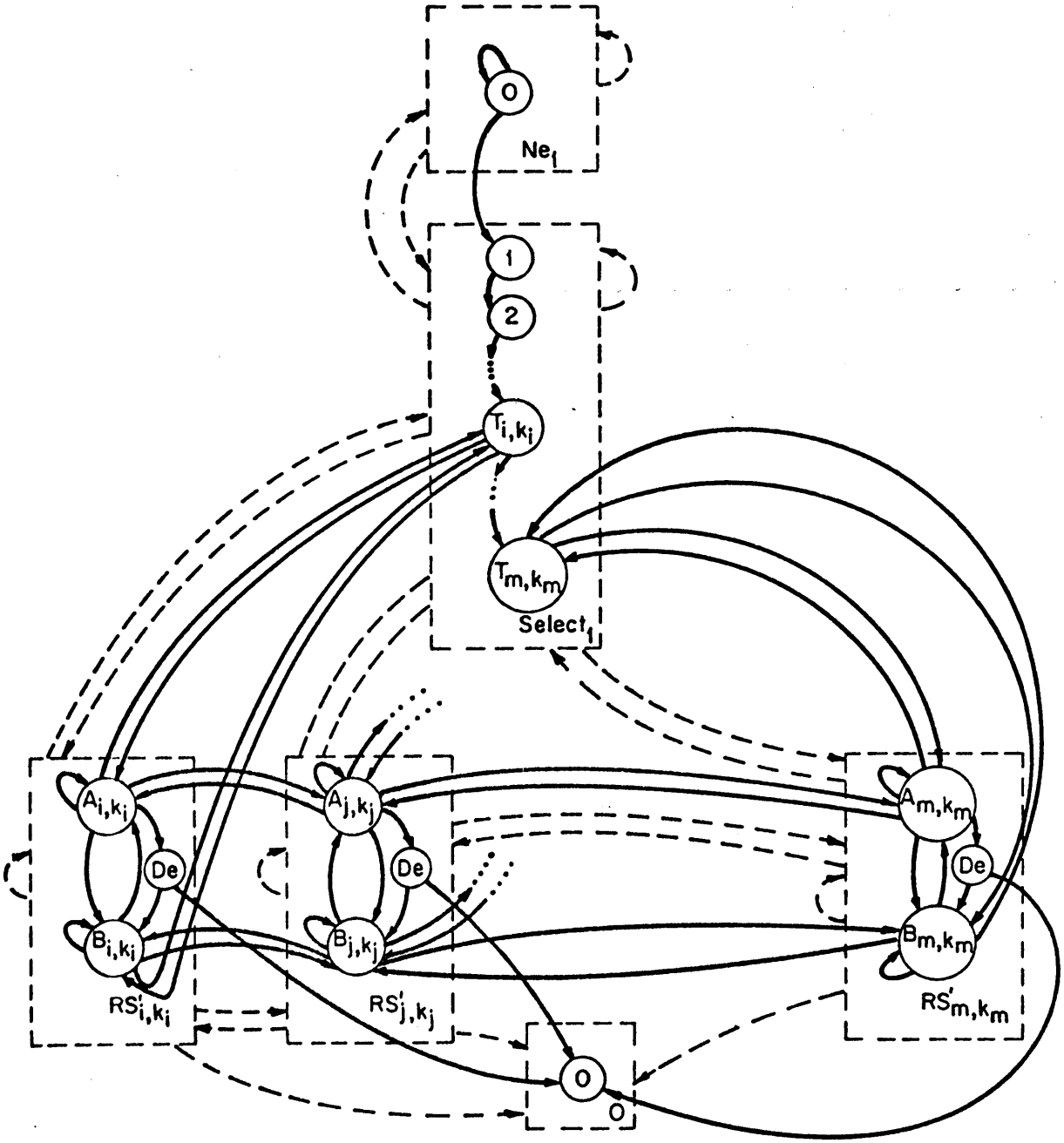
Now we describe the state structure of protein elements. Proteins, as discussed in Chapter 2, have the operational purpose of modulating[1]

[1]By modulation of a reaction system we understand the caralytic action of the protein. We assume that the speed of any reaction is zero in the absence of its catalytic protein. This assumption is justified by the fact that any organic reaction experiments an increase of several orders of magnitude (i.e., sometimes by a factor of 10,000) in its reaction speed in the presence of the appropriate catalytic molecule.



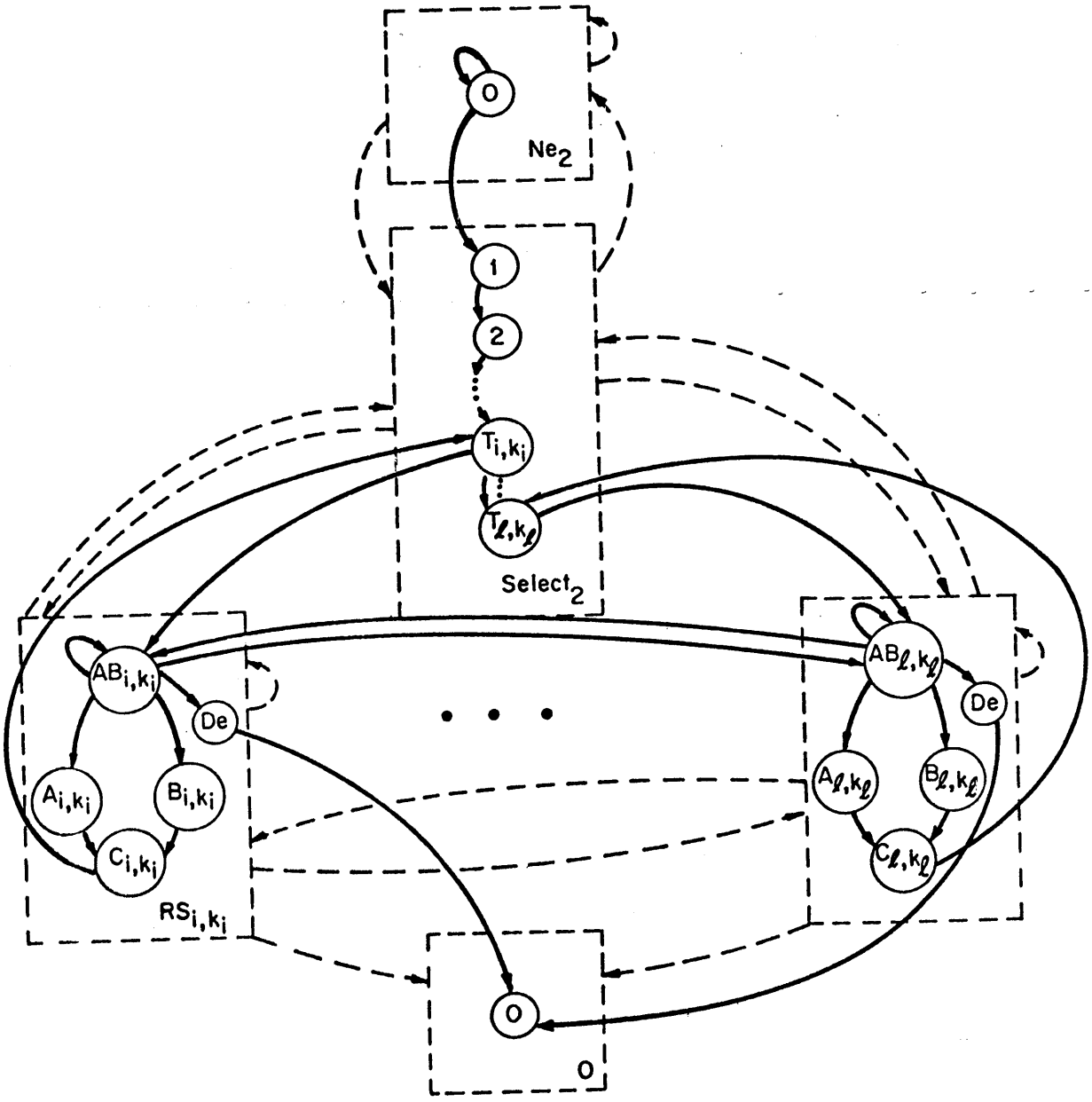
State Diagram of Protein Element of Irreversible Type (monomolecular reactions only)

Figure 10a



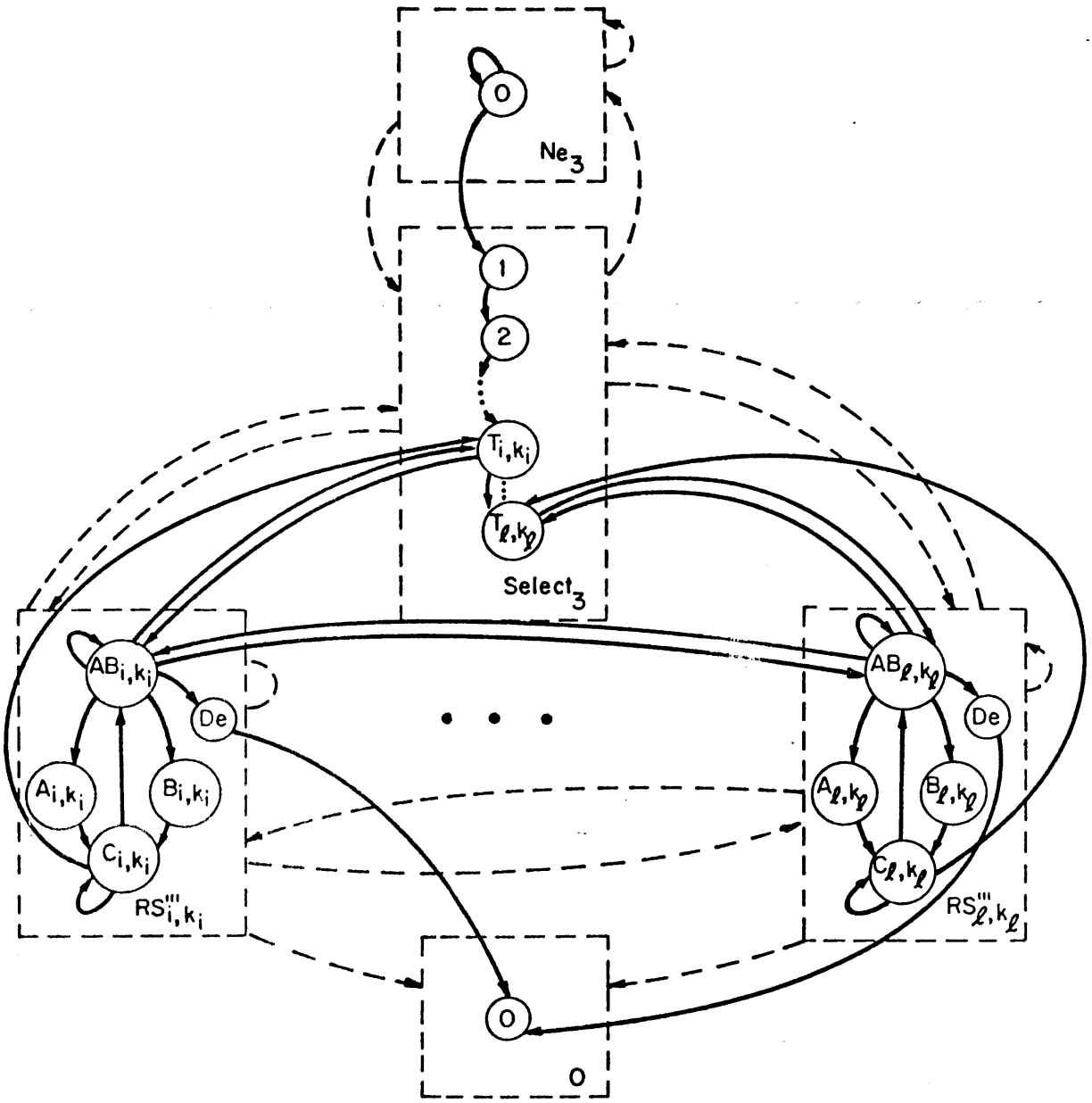
State Diagram of Protein Element of Reversible Type (monomolecular reactions)

Figure 10b



State Diagram of Protein Element of Irreversible Type (bimolecular reactions only)

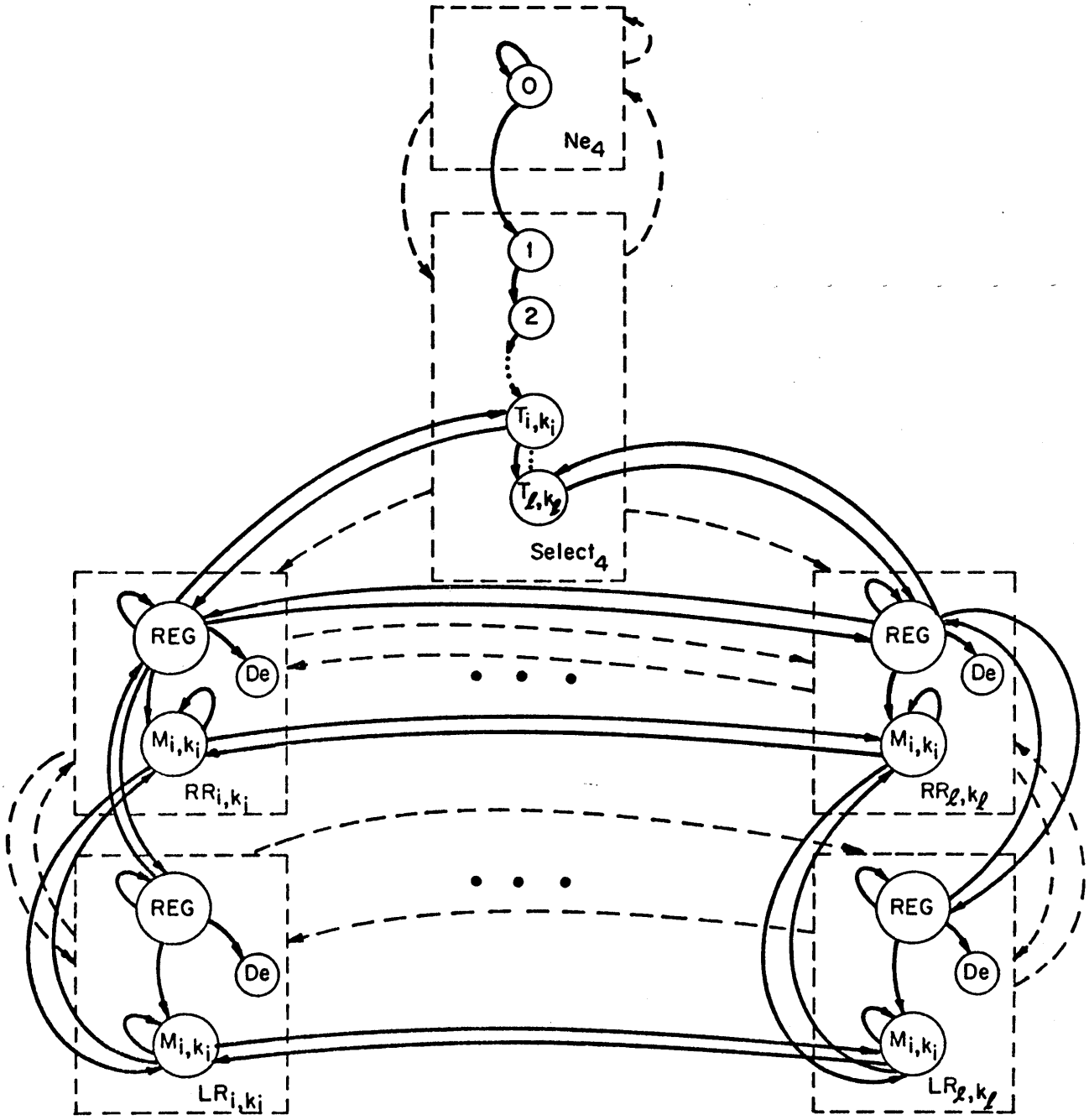
Figure 10c



State Diagram of Protein Element of Reversible Type

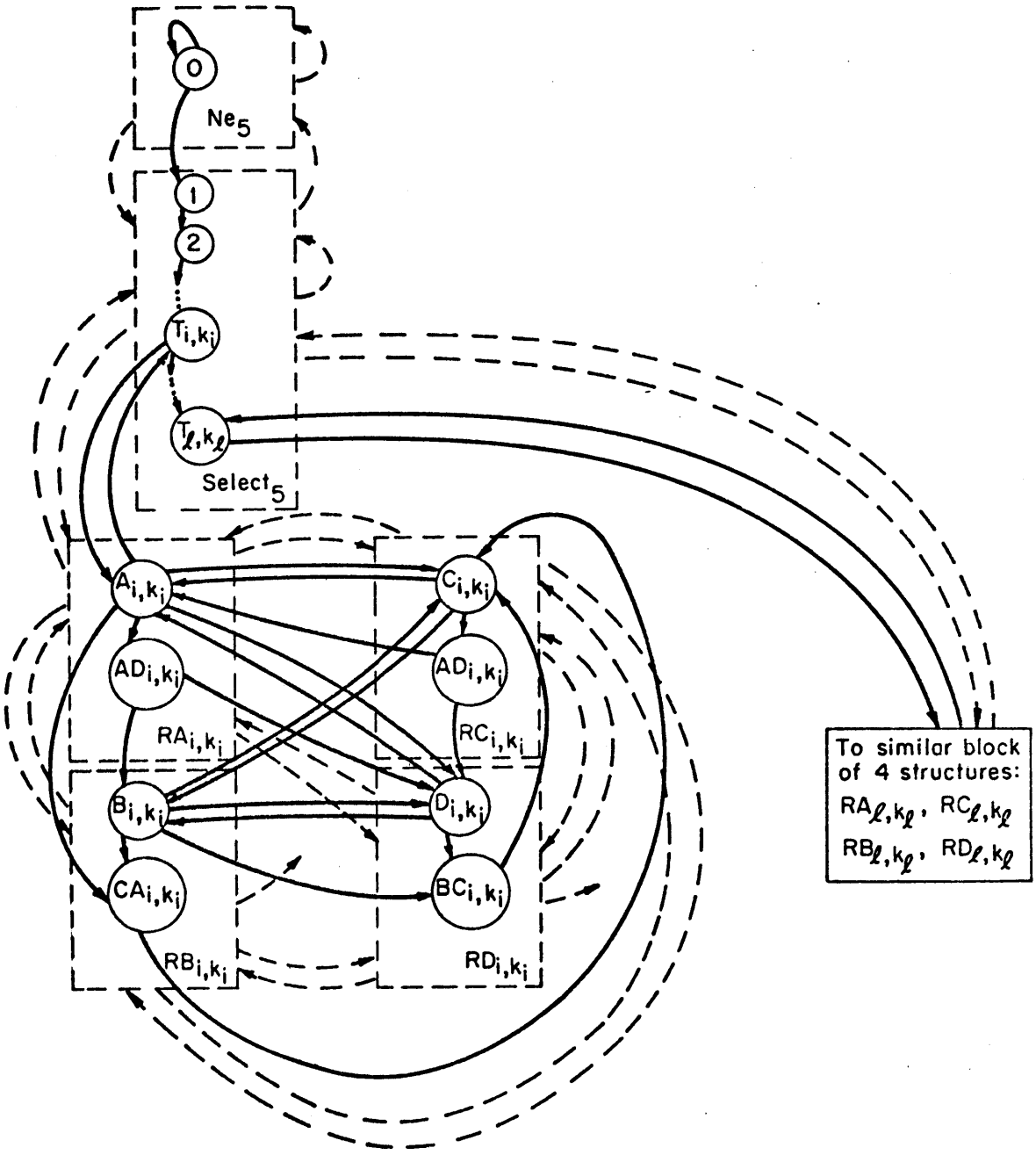
(bimolecular reactions only)

Figure 10d



State Diagram of Protein Element of Regulator Type

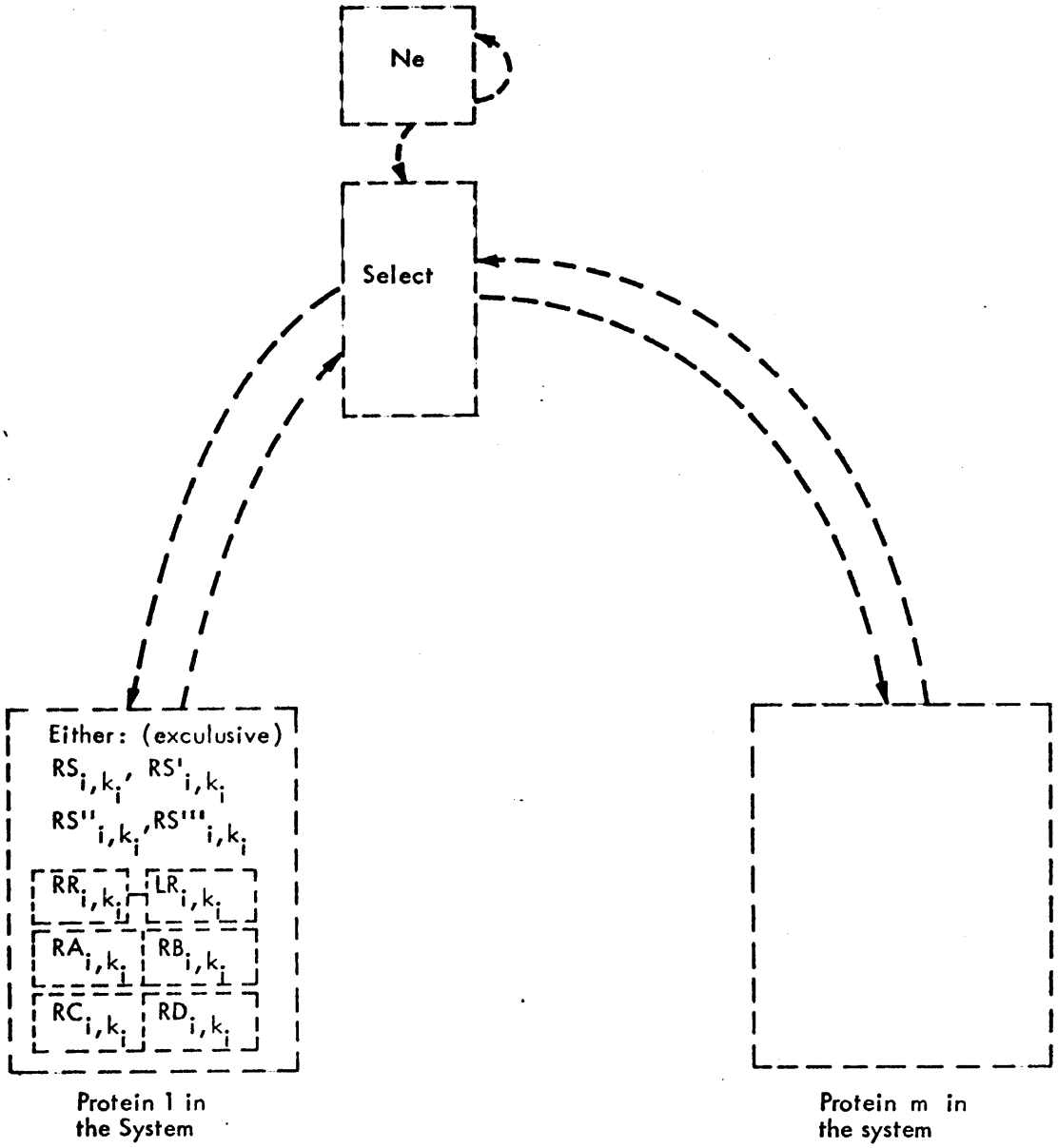
Figure 10e



State Diagram for Protein Element of Cycle Type

(see Figure 2.1-4(e))

Figure 10f



Illustrative Diagram of the Protein Type of a System with m Proteins (see discussion)

Figure 11

reaction systems, or to serve as regulatory elements for activating or deactivating transcription feedback mechanisms (TFM's) (see Section 2.1). In our model, we consider 5 different classes of reaction systems and one type of regulator protein, therefore we require 6 different types of protein elements. The state structure of these 6 types is developed in the following paragraphs.

Before we develop the state structure of the 6 types of protein elements to be considered in this study we discuss briefly an idealization about their synthesis and catalytic operational characteristics that has been made in order to simplify their state structure. It is well known (see Sampson [12]) that proteins start their catalytic activity while still under synthesis (i.e., a portion of the chain under synthesis has already the ability of performing the catalytic role of the complete chain[1]). However, in our model, the protein elements are assumed to be inactive as catalytic or modulating agents until the sunthesis has been completed. This assumption implies that the rates of the reaction systems controller by these proteins are actually higher than the ones that can be obtained from a simulation with the model[2]; but since reaction rates (or reaction constants) are not parameters used in our state formulation; this assumption does not seem to be critical.

---

[1] See Lewin [11], for a discussion on the experimental evidence supporting this fact.

[2] In Chapter 5 we will give a procedure for estimating reaction rates of a system on the basis of a simulation of its dynamics with our model.

The assumption described in the last paragraph, is needed for maintaining consistency in the level of aggregation of the state structure of protein elements with respect to the level of aggregation for the elements of the storage sub-class whose description was given earlier in this section. An analysis of the level of aggregation in the model, and its implications is given in Section 3.4.

State diagrams for each of the six classes of proteins we consider in our model are given in Figures 10-a through 10-f. The first 5 figures correspond to proteins able to catalyze sets of one of the 5 canonical reaction systems described in Section 2.1 (see Figure 2.1-4 and companion discussion). In Figure 10-f a state diagram corresponding to protein elements of regulator type (see Section 2.1) is given. Finally, in Figure 11 we combine these six steps into a single element, which is the type we use in our model for representing elements of the protein subclass in out model.

Figure 10-a gives the state representation of a protein element capable of representing one of m (m>0) proteins of the type that controls monomolecular irreversible reactions that is reactions of the form

$$A_{\ell,k_\ell} \xrightarrow[PR_{\ell,k_\ell}]{} B_{\ell,k_\ell} \qquad (3)$$

[1]The symbols A_{ℓ,k_ℓ} in (3) represent molecules. We will use these symbols also to certain intensities of protein and chemical elements, as discussed next.

where A_{ℓ,k_ℓ} is the reactant and B_{ℓ,k_ℓ} is the product of the k_ℓ-th reaction system associated with the ℓ-th operon of the system under study (see Section 2.1) and PR_{ℓ,k_ℓ} is the protein coded for by the k_ℓ-gene of the ℓ-th operon of the system.

Protein elements of the type described above have 4 classes of structure transition functions: N_e, Select, $\{RS_{i,k_i}\}$ and the dormant structure. In N_e the protein is in an inactive status (representing the availability of amino acid residues for conforming any of a finite number of protein elements all of them being catalytic agents in irreversible monomolecular reactions). In Select, the protein is under synthesis, and as discussed previoulsy, unable to catalyze any reactions.

In structure RS_{i,k_i}, the protein element represents the protein PR_{i,k_i} (i.e., the k_i-th protein coded for by the ith operon (OP_i), which is assumed to catalyze the monomolecular irreversible reactions $A_{i,k_i} \xrightarrow[PR_{i,k_i}]{} B_{i,k_i}$). Thus, the protein element is capable of representing one of m[1] different proteins. The set of structures RS_{i,k_i} are also involved in the transport of protein elements from the synthesis area in the informational space ($z_3 \equiv 2$, $z_2 \equiv -1$; see Figure 9) to the catalysis area (i.e., a region in the informational space where interaction among protein and chemical elements occur. This region will be

[1] m is an upper bound of the number of reaction systems in a hypothetical system consisting only of monomolecular reaction systems.

defined shortly), or to the transcriptional control area, if the protein element represents a regulator protein. The dormant structure 0, represents the destruction of the protein as a catalytic element.

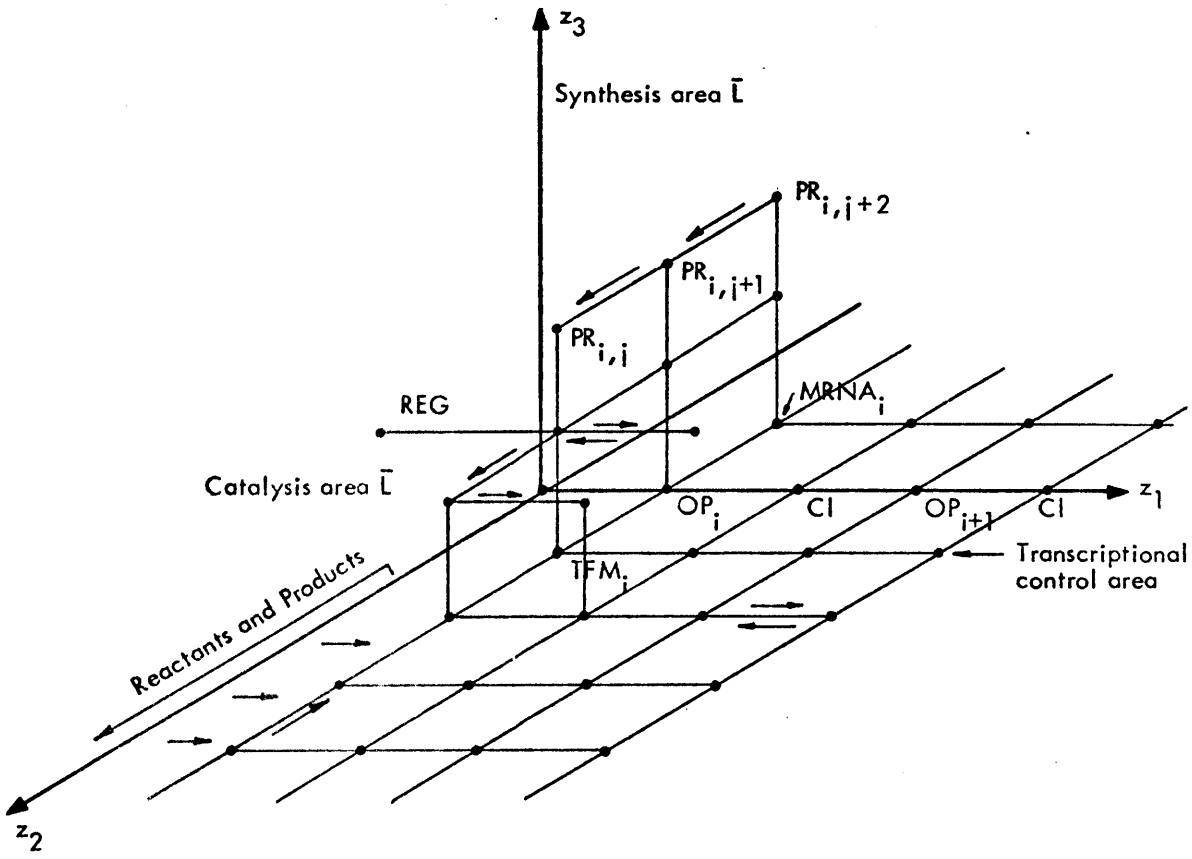
Before passing to discuss in detail the state of structure of the protein element sketched above, we will give a brief description of protein elements of the second class, (i.e., proteins catalyzing monomolecular reversible reactions). These two classes of elements have similar state structure and we will discuss their characteristics simultaneously.

A state diagram for this type is shown in Figure 10-b. As in the case of monomolecular irreversible protein types, the state structures have 4 kinds of structures labelled N_e, $Select_1$, RS'_{i,k_i} and 0 (the dormant structure). The operational purpose of these structures are analogous to the corresponding structures of monomolecular irreversible protein elements. The protein elements of the second class modulate one of a finite set of reactions of the form

$$A_{i,k_i} \underset{PR_{i,k_i}}{\overset{\rightarrow}{\leftarrow}} B_{i,k_i} \qquad (4)$$

Tables 12 and 13 give the structure and intensity transition functions for protein elements of the two classes sketched above. Figure 12 illustrates the motion of the protein elements in the informational space.

When the monomolecular reversible (resp. irreversible) protein type is in state (N_e, 0) (resp. (N_e, 0)) and the ribosome element interacting with it is in intensity cont (with the element allocation depicted in



Illustrative Diagram of Protein Motion (see text for explanation)

Informational Space

Figure 12

Table 12

Structure Transition Function for Protein Elements of Irreversible (Reversible) Monomilecular Type

Present Structure (Protein)	Present Intensity (Protein)	Present Intensity (Ribosome)	Present Intensity (Reactant Molecule)	Present Intensity (Product Molecule)	Next Structure (Protein)
Ne (Ne_1)	0	cont	--	--	select($select_1$)
Ne (Ne_1)	0	any≠cont	--	--	Ne(Ne_1)
select($select_1$)	$j \geq T_i, k_i$	cont	--	--	select($select_1$)
select($select_1$)	$j = T_i, k_i$	Br	--	--	$RS_i, k_i(RS'_i, k_i)$
select($select_1$)	j	Decont	--	--	Ne(Ne_1)
$RS_i, k_i(RS'_i, k_i)$	A_i, k_i	--	A_i, k_i	--	$RS_i, k_i(RS'_i, k_i)$
select($select_1$)	T_i, k_i	--	A_i, k_i	--	$RS_i, k_i(RS'_i, k_i)$
select($select_1$)	T_i, k_i	--	--	(B_i, k_i)	-- (RS'_i, k_i)

Table 12 (contd.)

Protein Motion

Present Structure (Protein)	Present Intensity							Next Structure (Protein)
	Neighbor 0	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
$RS_i,k_i(RS_i',k_i)$	A_i,k_i	any	0	any	$k_i' \neq k_i$ A_i,k_i	any	M	0(0)
$RS_i,k_i(RS_i',k_i)$	A_i,k_i	any	$k_i' \neq k_i$ A_i,k_i'	any	$k_i'' \neq k_i' = k_i$ 0 or A_i,k_i''	any	any	RS_i,k_i' (RS_i',k_i)
$RS_i,k_i(RS_i',k_i)$	A_i,k_i	any	any	REG	any	$k_i \neq k_i'$ A_i,k_i	any	RS_i,k_i (RS_i',k_i)
$RS_i,k_i(RS_i',k_i)$	A_i,k_i $(A_i,k_i,$ $B_i,k_i)$	any	any	any	any	any	A_i,k_i $(A_i,k_i$ $B_i,k_i)$	select(select$_1$)
$RS_i,k_i(RS_i',k_i)$	A_i,k_i $(A_i,k_i,$ $B_i,k_i)$	any≠stop	any	$\ell \neq i$ A_ℓ,k_i $(A_\ell,k_\ell,B_\ell,k_i)$	any	any	$j \neq 1$ A_j,k_j (A_j,k_j,B_j,k_j)	RS_ℓ,k_ℓ (RS_ℓ',k_ℓ)
$RS_i,k_i(RS_i',k_i)$	$(A_i,k_i)(A_i,k_i,$ $B_ik_i)$	stop	any	$\ell \neq i$ A_ℓ,k_ℓ $(A_\ell,k_\ell,B_\ell,k_\ell)$	any	any	stop	0(0)
$RS_i,k_i(RS_i',k_i)$	De	any	any	any	any	any	stop	0(0)

Table 13

Intensity Transition Functions for Protein Elements of Irreducible (Reducible) Monomolecular Type

Ne (Ne_1)

Present Intensity (Protein)	Present Intensity (Ribosome)	Next Intensity (Protein)
0	cont	1
0	any≠cont	0

select ($select_1$)

Present Intensity (Protein)	Present Intensity (Ribosome)	Next Intensity (Protein)
j j=1,2,...	cont	j+1
$j=T_i,k_i$	Br	A_i,k_i
j	Decont	0

Table 13 (contd.)

Present Intensity (Protein)	Present Intensity						Next Intensity (Protein)
	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
A_i,k_i	--	--	--	--	--	A_i,k_i (A_i,k_i)	B_i,k_i (B_i,k_i)
A_i,k_i	--	--	--	--	--	any $\neq (B_i,k_i)$ A_i,k_i (A_i,k_i)	A_i,k_i (A_i,k_i)
A_i,k_i	--	0	--	--	--	--	0
A_i,k_i	--	$k_i'=k_i$ A_i,k_i	--	--	--	--	A_i,k_i' (A_i,k_i')
A_i,k_i	--	--	REG	--	A_i,k_i	--	A_i,k_i' (A_i,k_i')
B_i,k_i	--	--	--	--	--	(B_i,k_i)	(A_i,k_i)
A_i,k_i	--	--	A_j,k_j	--	--	--	A_j,k_j (A_j,k_j)
A_i,k_i	stop	--	A_j,k_j	--	--	stop	De (De)
A_i,k_i	A_j,k_j	--	stop	--	--	$A,k \neq i$	A_j,k_j (A_j,k_j)
A_i,k_i	--	--	stop	--	--	stop	De (De)
De	--	--	stop	--	--	stop	0 (0)
B_i,k_i	--	--	--	--	--	B_i,k_i	T_i,k_i (A_i,k_i)

Figure 9), the protein element suffers a state transition to state (Select, 1) (resp. ($Select_1$, 1)). If the intensity of the ribosome elements is still cont, the corresponding protein element suffers successive state transitions to (Select, 2) (resp. ($Select_1$, 2)), (Select, 3) (resp. ($Select_1$, 3), and so on until, for some j, $j=T_{i,k_i}$ the time of synthesis of and of the proteins simulated by the element. This time of synthesis, as discussed previously, is detected by the protein element when the corresponding ribosome reaches intensity Br. At this time, the protein suffers a state transition from (Select, T_{i,k_i}) (resp. ($Select_1$, T_{i,k_i})) to state (RS_{i,k_i}, A_{i,k_i}) (resp. (RS'_{i,k_i}, A_{i,k_i})).

When in structure RS_{i,k_i} (resp. RS'_{i,k_i}) and intensity A_{i,k_i} (resp. (A_{i,k_i} or B_{i,k_i}) the protein element attains one of several states depending on whether it is presently located in the synthesis or catalysis[1] areas of the informational space (these areas are indicated in the diagram of Figure 12). In the catalysis area, (the points of the informational space with loci $z_2 = 2$, $z_3 = 1$), if the intensity of the corresponding nearest-neighbor with $z_2 = 2$, $z_3 = 0$ and the same z_1-ordinate as the protein element is A_{i,k_i} the intensity of the protein element changes to B_{i,k_i} conversely, if the intensity of the reversible protein

[1]We note that the protein elements "knows" its position in the several regions (synthesis or catalysis areas or the path in between) of the information space by the intensities of its nearest neighbors as explained above.

element is B_{i,k_i} and the intensity of the element at $z_2 = 2$, $z_3 = 0$ and the same z_1-ordinate as the protein element is B_{i,k_i} then the protein element changes to A_{i,k_i}.

If the intensity of the nearest neighbor at $z_2 = 0$, $z_3 = 0$ and some z_1 of a protein element in the catalysis area is any other than A_{i,k_i}, say A_{ℓ,k_ℓ} (representing the chemical element A_{ℓ,k_ℓ}) the protein intensity changes to that of its nearest 3-neighbor (see Table 13) while that of its nearest 1-neighbor becomes A_{i,k_i}; that is, the protein element is <u>transported</u> one unit in the positive direction along the corresponding z_1-ordinate, ($z_2 \equiv 2$, $z_3 \equiv 1$). This translation ends if and when the protein element interacts with a chemical element in the correct intensity (i.e., A_{i,k_i}), allocated in its 5-neighbor, $z_2 = 2$, $z_3 = 0$ and some z_1, or it finds an element in state (STOP, STOP) in that location, as indicated in Table 13.

When in structure RS_{i,k_i} (resp. RS'_{i,k_i}) and intensity A_{i,k_i} and, the protein is in the synthesis area (see Figure 12) and its 2- and 4-neighbors are in intensity 0 or the intensity of the 4-neighbor is $A_{i,k_{i-1}} = A_{i,k'_i}$ (i.e., the intensity corresponding to the previously synthesized protein of OP_i), the intensity of the protein element suffers an intensity transition to 0 while that of the 4-neighbor becomes A_{i,k_i}. Thus, effectively, the protein element has effected a <u>translation</u> in the positive direction of z_2 (i.e. it is now allocated on $z_2 = 0$, $z_3 = 2$ and the same z_1-ordinate as that of OP_i, as shown in Figure 12).

When the protein element is in state (RS_{i,k_i}, A_{i,k_i}) (resp. (RS'_{i,k_i}, A_{i,k_i})) and allocated to the point in the informational space $z_2 = 1$, $z_3 = 1$ and the same z_1-ordinate as TFM_i (see Figure 12) and its 5-neighbor is in intensity A_{i,k_i}, the protein element suffers a state transition from (RS_{i,k_i}, A_{i,k_i}) (resp. (RS'_{i,k_i}, A_{i,k_i})) to (RS'_{i,k_i}, A_{i,k_i}) (resp. $(RS'_{i,k_i}, A_{i,k'_i})$).[1]

We note that the state transition steps described in the last 3 paragraphs represent the translation of a protein element from the synthesis to the catalysis area (a procedure that takes 3 time steps); once there, the state dynamics of the protein element has the structure described previously.

We note that the "distance"[2] of transport of a protein element between the synthesis and catalysis areas in our model is 3 and it is the same <u>for all protein elements</u>. In the cell, this is not the case in general since the sites at which reactions occur are scattered all over the intracellular space or affixed to the membrane walls and the distances the proteins must travel from the synthesis area to the places where they perform their catalytic actions, are widely different from each other.

We could allocate in the <u>reactants-and-products region</u> (see Fig-

---

[1]Notice that in this transport step the protein element is translating (one step) in the negative direction of the z_3-axis.

[2]By distance here, we mean the number of points in the informational space occupied by the protein element PR_{i,k_i} in state (RS_{i,k_i}, A_{i,k_i}) (resp. (RS'_{i,k_i}, A_{i,k_i}).

ure 12) the reactants of different reaction systems at distances from the corresponding synthesis points so as to reflect the actual distances in the cell; however, for most reaction systems, these distances are not known. Thus, we have assumed with our reactant/product element allocation, an equal distance for all the systems in the model.

Now we describe the state structure for protein elements that catalyze bimolecular irreversible (resp. reversible) reactions. A state transition diagram for the bimolecular irreversible (resp. reversible) protein elements is given in Figure 10-c (resp. 10-d). The corresponding structure and transition functions are given in Tables 14 and 15, respectively.

The reactions in which these proteins perform catalytic actions are of the form

$$A_{i,k_i} + B_{i,k_i} \xrightarrow[PR_{i,k_i}]{} C_{i,k_i} \quad \text{(irreversible)} \tag{5}$$

$$A_{i,k_i} + B_{i,k_i} \underset{PR_{i,k_i}}{\overset{\rightarrow}{\leftarrow}} C_{i,k_i} \quad \text{(reversible)} \tag{6}$$

where A_{i,k_i}, B_{i,k_i} are the reactants[1] and C_{i,k_i} is the product of the k_i-th reaction system associated with the i-th operon.

---

[1] As in the monomolecular case, we use the symbols A_{i,k_i}, B_{i,k_i} and C_{i,k_i} to indicate molecules as well as intensities of protein and chemical elements. The meaning intended is made explicit in each case in which these symbols are used.

Table 14

Structure Transition Function for Protein Elements of Irreversible (Reversible) Bimolecular Type

Present Structure (Protein)	Present Intensity (Protein)	Present Intensity (Ribosome)	Present Intensity			Next Structure (Protein)
			(1st Reactant Molecule)	(2nd Reactant Molecule)	(Product Molecule)	
$Ne_2(Ne_3)$	0	cont	--	--	--	$select_2(select_3)$
$Ne_2(Ne_3)$	0	any $\neq$ cont	--	--	--	$Ne_2(Ne_3)$
$select_2(select_3)$	$j \neq \overline{T}_i,k_i$	cont	--	--	--	$select_2(select_3)$
$select_2(select_3)$	$j = T_i,k_i$	Br	--	--	--	RS''_i,k_i (RS'''_i,k_i)
$select_2(select_3)$	j	Decont	--	--	--	$(Ne_2)(Ne_3)$
RS''_i,k_i (RS'''_i,k_i)	AB_i,k_i	--	A_i,k_i	--	--	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (RS'''_i,k_i)	AB_i,k_i	--	B_i,k_i	--	--	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (RS'''_i,k_i)	AB_i,k_i	--	--	A_i,k_i	--	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (RS'''_i,k_i)	AB_i,k_i	--	--	B_i,k_i	--	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (RS'''_i,k_i)	A_i,k_i	--	B_i,k_i	--	--	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (RS'''_i,k_i)	A_i,k_i	--	--	B_i,k_i	--	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (RS'''_i,k_i)	B_i,k_i	--	--	A_i,k_i	--	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (RS'''_i,k_i)	B_i,k_i	--	A_i,k_i	--	--	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (RS'''_i,k_i)	C_i,k_i	--	--	--	C_i,k_i	$select_2(select_3)$
$select_2(select_3)$	T_i,k_i	--	A_i,k_i	--	--	RS''_i,k_i (RS'''_i,k_i)
			B_i,k_i	A_i,k_i		
$select_2(select_3)$	T_i,k_i	--	--	B_i,k_i	--	RS''_i,k_i (RS'''_i,k_i)
$select_2(select_3)$	T_i,k_i	--	--	--	C_i,k_i	-- (RS'''_i,k_i)

Table 14 (contd.)

Present Structure (Protein)	Present Intensity							Next Structure (Protein)
	Neighbor 0	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
RS''_i,k_i (RS'''_i,k_i)	AB_i,k_i	any	0	any	$k_i + k_i$ A_i,k_i	any	Br	0(0)
RS''_i,k_i (RS'''_i,k_i)	AB_i,k_i	any	$k'_i \neq k_i$ 0 or A_i,k_i	any	$k''_i \neq k'_i \neq k_i$ 0 or $A_i k_i$	any	any	RS''_i,k'_i (RS'''_i,k_i)
RS'''_i,k_i (R'''_i,k_i)	AB_i,k_i	any	any	REG	any	$k_i \neq k_i$ AB_i,k_i	any	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (R'''_i,k_i)	AB_i,k_i	any	$\ell \neq i$ AB_ℓ,k_ℓ	any	any	$m \neq i$ A_m,k_m	any	RS''_ℓ,k_ℓ (RS'''_ℓ,k_ℓ)
RS''_i,k_i (R'''_i,k_i)	AB_i,k_i	any	AB_ℓ,k_ℓ	any	any	$m \neq i$ B_m,k_m	any	RS''_ℓ,k_ℓ (RS'''_ℓ,k_ℓ)
RS''_i,k_i (RS'''_i,k_i)	A_i,k_i, B_i,k_i AB_i,k_i	any	any	$A_\ell,k_\ell, B_\ell,k_\ell$ $\neq i$ AB_ℓ,k	any	$m \neq i$ A_m,k_m	any	RS''_ℓ,k_ℓ (RS'''_ℓ,k_ℓ)
RS''_i,k_i (R'''_i,k_i)	A_i,k_i, B_i,k_i AB_i,k_i	stop	any	any	any	any	any	RS''_i,k_i (RS'''_i,k_i)
RS''_i,k_i (R'''_i,k_i)	De	stop	any	any	any	stop	stop	0(0)

Table 15

Intensity Transition Function for Protein Elements of Irreversible (Reversible) Bimolecular Type

Ne_2 (Ne_3)

Present Intensity (Protein)	Present Intensity (Ribosome)	Next Intensity (Protein)
0	cont	1
0	any≠cont	0

$Select_2$ ($Select_3$)

Present Intensity (Protein)	Present Intensity (Ribosome)	Present Intensity (Reactant Molecules)	Present Intensity (Product Molecule)	Next Intensity (Protein)
j_1 $j \neq T_i, k_i$	cont	--	--	j+1
$j = T_i, k_i$	Br	--	--	AB_i, k_i
T_i, k_i	--	A_i, k_i, B_i, k_i	--	B_i, k_i
T_i, k_i	--	B_i, k_i	--	A_i, k_i
T_i, k_i	--	--	C_i, k_i	(AB_i, k_i)

Table 15 (contd.)

Present Intensity (Protein)	Present Intensity						Next Intensity (Protein)
	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
AB_i,k_i	--	--	--	--	--	A_i,k_i	B_i,k_i
AB_i,k_i	--	--	--	--	--	B_i,k_i	A_i,k_i
A_i,k_i	--	--	--	--	--	A_i,k_i	C_i,k_i
B_i,k_i	--	--	--	--	--	B_i,k_i	C_i,k_i
C_i,k_i	--	--	--	--	--	A_i,k_i	T_i,k_i (AB_i,k_i)
AB_i,k_i	--	AB_i,k'_i	--	$k''_i \neq k'_i \neq k_i$ 0 or AB_i,k''_i	--	--	AB_i,k'_i
AB_i,k_i	--	0	--	$k_i \neq k'_i$ A_i,k_i	--	M	0
AB_i,k_i	--	--	REG	--	$k_i \neq k'_i$ AB_i,k_i	--	AB_i,k_i
AB_i,k_i	--	$\ell+i$ AB_ℓ,k_ℓ	--	--	$m \neq i$ B_m,k_m	--	A_ℓ,k_ℓ
AB_i,k_i,A_i,k_i,B_i,k_i	--	--	$A_\ell,k_\ell,B_\ell,k_\ell$ $\ell \neq i$ AB_ℓ,k_ℓ	--	$m \neq i$ A_m,k_m	--	$A_\ell,k_\ell,B_\ell,k_\ell$ AB_ℓ,k_ℓ
AB_i,k_i,A_i,k_i,B_i,k_i	stop	--	--	--	--	--	De
De	stop	--	--	--	--	stop	0

The protein elements whose state structure is of the form shown in Figure (10-c) resp. (10-d) can catalyze one of a finite set of biomolecular irreversible (resp. reversible) reactions.

The state structure of protein element of bimolecular irreversible (resp. reversible) type is composed of kinds of structures: Ne_2, $Select_2$, $\{RS''_{i,k_i}\}$ and the dormant structure 0 (resp. Ne_3, $Select_3$, $\{RS'''_{i,k_i}\}$ and 0).

When in state (Ne_2, 0) (resp. (Ne_3, 0)) the protein element is in an inactive status, allocated in the synthesis area (see Fig. 12) and waiting for protein synthesis to start. Synthesis starts, when the intensity of the ribosome allocated in the 5-nearest neighboring point of the protein element becomes cont, the protein element's state becomes ($select_2$, 1) (resp. ($select_3$,1)); if the intensity of the ribosome is still cont, a transition to ($select_2$, 2) (resp.($select_3$, 2)); if the ribosomes intensity is still cont the protein state becomes ($select_2$, 3) (resp. ($select_3$, 3) and so on, until for some intensity j, $j=t_{i,k_i}$ the time of synthesis of one of the preteins represented by the protein element, and the intensity of the ribosome becomes M. In such a condition, the protein element suffers a state transition to (RS''_{i,k_i}, AB_{i,k_i}) (resp. RS'''_{i,k_i}, AB_{i,k_i})). One in state (RS''_{i,k_i}, AB_{i,k_i}), the protein element is transported from the synthesis to the catalysis region in a manner analogous to the transfer procedure described earlier for protein elements of monomolecular i-reversible (resp. reversible) type.

Once in the catalysis region, the protein element in state (RS''_{i,k_i}, AB_{i,k_i}) suffers one of several possible state transitions. If the element allocated in its 5-nearest neighboring location is in A_{i,k_i} intensity the protein element suffers a state transition to (RS''_{i,k_i}, B_{i,k_i}), if this element is in intensity B_{i,k_i} then the protein elements state becomes (RS''_{i,k_i}, A_{i,k_i}). An analogous state transition occurs in the reversible case when the protein element is in state (RS''_{i,k_i}, AB_{i,k_i}).

If the protein element state in the catalysis region is either a) (RS''_{i,k_i}, AB_{i,k_i}), b) (RS''_{i,k_i}, A_{i,k_i}) or, c) (RS''_{i,k_i}, B_{i,k_i}) and the intensity of its 5-nearest neighbor is different from either A_{i,k_i} or B_{i,k_i} for a), A_{i,k_i} for b) or B_{i,k_i} for c) and its 3-nearest neighbor is in intensity AB_{ℓ,k_ℓ} or A_{ℓ,k_ℓ} or B_{ℓ,k_ℓ}, the state of the protein element becomes, respectively, (RS_{ℓ,k_ℓ}, AB_{ℓ,k_ℓ}) or (RS''_{ℓ,k_ℓ}, A_{ℓ,k_ℓ}) or (RS''_{ℓ,k_ℓ}, B_{ℓ,k_ℓ}). Thus, effectively, a transfer of the protein element in the positive direction of the z_1-axis has occured. This translation continues until an element on the 5-nearest neighboring location is of the appropriate intensity (i.e., AB_{i,k_i} or A_{i,k_i} or B_{i,k_i}) or this intensity is stop, in which case the proteins elements state becomes (0, 0).

Finally, if the state of the protein element in the catalysis region is (RS''_{i,k_i}, A_{i,k_i}) or RS''_{i,k_i}, B_{i,k_i}), and the corresponding 5-nearest neighboring element is A_{i,k_i} or B_{i,k_i}, respectively, the state of the element becomes (RS''_{i,k_i}, C_{i,k_i}) (resp. (RS'''_{i,k_i}, C_{i,k_i})) indicating the completion of the catalysis action for one irreversible

(resp. reversible) bimolecular reaction system. In the irreversible case, the state then becomes (select$_2$, T_{i,k_i}) in the reversible case it then becomes (RS''_{i,k_i}, AB_{i,k_i}).

We note that the catalysis action described above, proceeds always in two steps:

(1) If the protein element in intensity AB_{i,k_i} interacts with an element in intensity A_{i,k_i}, (or B_{i,k_i}) in its 5-nearest neighbor location, then its intensity becomes B_{i,k_i}, (or A_{i,k_i}).

(2) If the protein element in intensity B_{i,k_i}, (or A_{i,k_i}) interacts with an element in intensity B_{i,k_i}, (or A_{i,k_i}) the reaction is completed as indicated proviously, otherwise the protein element moves along the catalysis region until such an element, if ever, is found or the protein element interacts with one STOP element.

Before discussing the state structure of the last two types of protein elements we digress for a moment to describe the state structure of the genetically controller transformation elements (see Section 3.2) which represent reaction systems, each one controller by one of the 4 protein element types discussed in the last paragraphs.

The state diagram corresponding to transformation elements under catalytic control of proteins of monomolecular type (both reversible and irreversible) is shown in Figure 13. The corresponding structure and intensity transition functions are given in Tables 16 and 17 respectively.

Table 16

Structure Transition Function of the Transformation Monomolecular Element

Present Structure (Reaction System)	Present Intensity					Next Structure (Reaction System)
	(Neighbor 0)	(Neighbor 1)	(Neighbor 4)	(Neighbor 3)	(Neighbor 5)	
Rr_i,k_i	A_i,k_i	--	--	--	A_i,k_i	Rr_i,k_i
$R\ell_i,k_i$	A_i,k_i	--	--	--	A_i,k_i	$R\ell_i,k_i$
Rr_i,k_i	B_i,k_i	--	--	--	B_i,k_i	$-(R\ell_i,k_i)$
$R\ell_i,k_i$	B_i,k_i	--	--	--	B_i,k_i	$-(R\ell_i,k_i)$
Rr_i,k_i	B_i,k_i A_i,k_i	--	B_ℓ,k_ℓ A_ℓ,k_ℓ	--	0	Rr_ℓ,k_ℓ
$R\ell_i,k_i$	B_i,k_i A_i,k_i	--	B_ℓ,k_ℓ A_ℓ,k_ℓ	--	0	$R\ell_\ell,k_\ell$
Rr_i,k_i	B_i,k_i A_i,k_i	$\neq$ stop	--	B_ℓ,k_ℓ A_ℓ,k_ℓ	$\neq B_i,k_i$ $\neq A_i,k_i$	Rr_ℓ,k_ℓ
Rr_i,k_i	B_i,k_i A_i,k_i	stop	--	--	--	$R\ell_i,k_i$
$R\ell_i,k_i$	B_i,k_i A_i,k_i	B_ℓ,k_ℓ A_ℓ,k_ℓ	--	$\neq$ stop	$\neq B_i,k_i$ $\neq A_i,k_i$	$R\ell_\ell,k_\ell$
R_i,k_i	B_i,k_i A_i,k_i	--	--	stop	--	Rr_i,k_i
Rr_i,k_i	B_i,k_i A_i,k_i	stop	--	--	stop	0
R_i,k_i	B_i,k_i A_i,k_i	--	--	stop	stop	0

Table 17

Intensity Transition Functions of the Transformation

Monomolecular Element $A_i,k_i \rightarrow B_i,k_i$

$A_i,k_i \overset{\rightarrow}{\leftarrow} B_i,k_i$

Rr_i,k_i

Present Intensity (Reaction System)	Present Intensity				Next Intensity (Reaction System)
	Neighbor 1	Neighbor 4	Neighbor 3	Neighbor 5	
A_i,k_i	--	--	--	A_i,k_i	B_i,k_i
B_i,k_i	--	--	--	B_i,k_i	$0(A_i,k_i)$[1]
A_i,k_i	--	A_ℓ,k_ℓ[2]	--	0	A_ℓ,k_ℓ
A_i,k_i	--	B_ℓ,k_ℓ	--	0	B_ℓ,k_ℓ
B_i,k_i	--	B_ℓ,k_ℓ	--	0	B_ℓ,k_ℓ
B_i,k_i	--	A_ℓ,k_ℓ	--	0	A_ℓ,k_ℓ
A_i,k_i	stop	--	--	--	A_i,k_i
B_i,k_i	stop	--	--	--	B_i,k_i
A_i,k_i	any≠stop	--		A_ℓ,k_ℓ $\ell\neq i$	A_ℓ,k_ℓ
A_i,k_i	any≠stop	--	B_ℓ,k_ℓ $\ell\neq i$	--	B_ℓ,k_ℓ
B_i,k_i	any≠stop	--	B_ℓ,k_ℓ $\ell\neq i$	--	B_ℓ,k_ℓ
B_i,k_i	any≠stop	--	A_ℓ,k_ℓ $\ell\neq i$	--	A_ℓ,k_ℓ

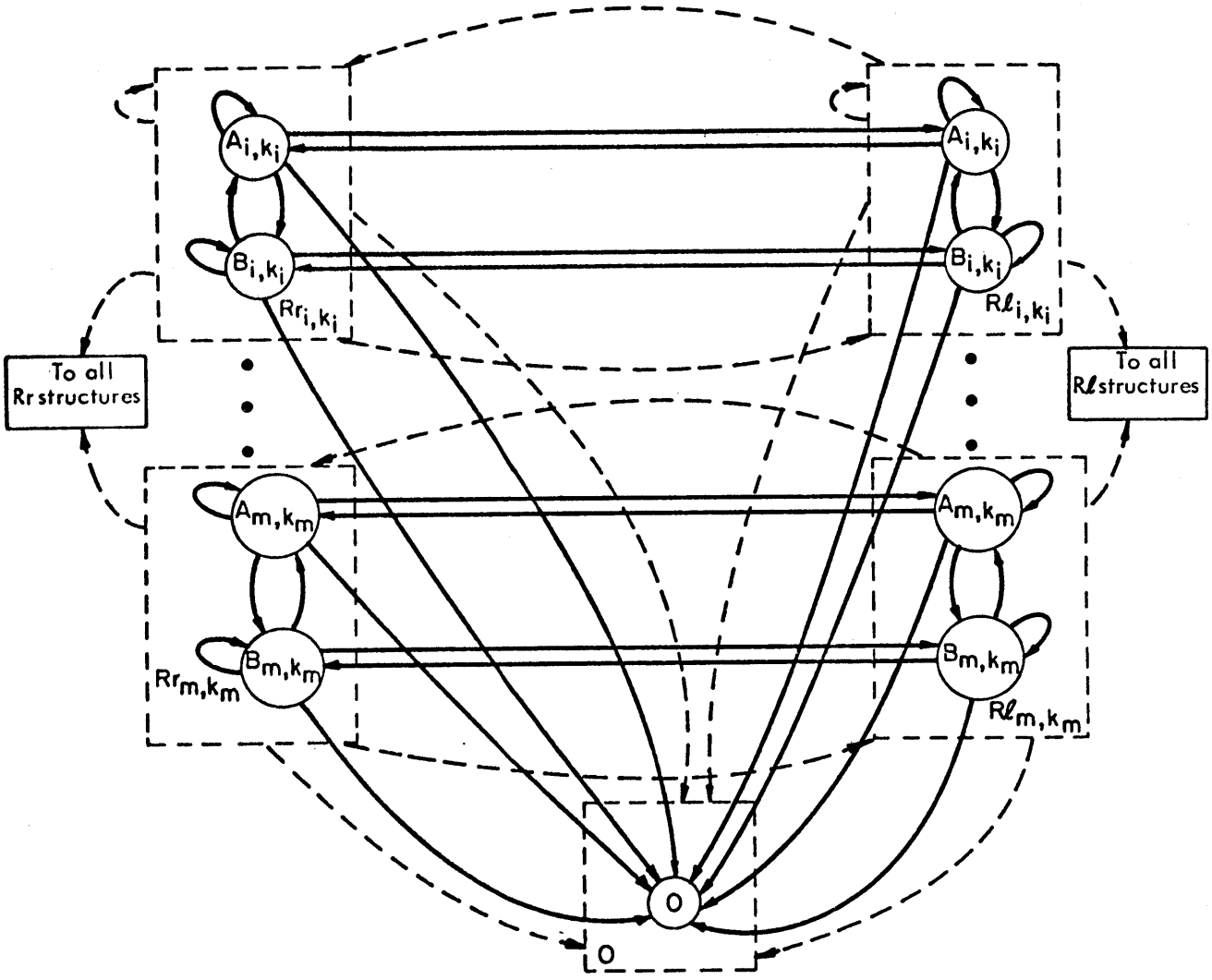
[1]The intensity in parenthesis is reached if the corresponding protein is of reversible type.

[2]Not necessarily different from i.

Table 17 (contd.)

$R\ell_i,k_i$

Present Intensity (Reaction System)	Present Intensity				Next Intensity (Reaction System)
	Neighbor 3	Neighbor 4	Neighbor 1	Neighbor 5	
A_i,k_i	--	--	--	A_i,k_i	B_i,k_i
B_i,k_i	--	--	--	B_i,k_i	$0(A_i,k_i)$
A_i,k_i	--	A_ℓ,k_ℓ	--	0	A_ℓ,k_ℓ
A_i,k_i	--	B_ℓ,k_ℓ	--	0	B_ℓ,k_ℓ
B_i,k_i	--	B_ℓ,k_ℓ	--	0	B_ℓ,k_ℓ
B_i,k_i	--	A_ℓ,k_ℓ	--	0	A_ℓ,k_ℓ
A_i,k_i	stop	--	--	--	A_i,k_i
B_i,k_i	stop	--	--	--	B_i,k_i
A_i,k_i	any≠stop	--	A_ℓ,k_ℓ $\ell\neq i$	--	A_ℓ,k_ℓ
A_i,k_i	any≠stop	--	B_ℓ,k_ℓ $\ell\neq i$	--	B_ℓ,k_ℓ
B_i,k_i	any≠stop	--	B_ℓ,k_ℓ $\ell\neq i$	--	B_ℓ,k_ℓ
B_i,k_i	any≠stop	--	A_ℓ,k_ℓ $\ell\neq i$	--	A_ℓ,k_ℓ



[1]See text and Tables 16 and 17.

State Diagram of the Transformation Monomolecular Element

Figure 13

The state diagram corresponding to transformation elements under catalytic control of proteins of bimolecular type both, reversible and irreversible is shown in Figure 14. The corresponding structure and intensity transition functions are given in Tables 18 and 19, respectively.

The state structure of the type representing monomolecular transformation elements is composed of 3 classes of structures: $\{Rr_{i,k_i}\}$, $\{R\ell_{i,k_i}\}$, and the dormant structure.

Monomolecular transformation elements are allocated <u>initially</u> in the region of the informational space determined by

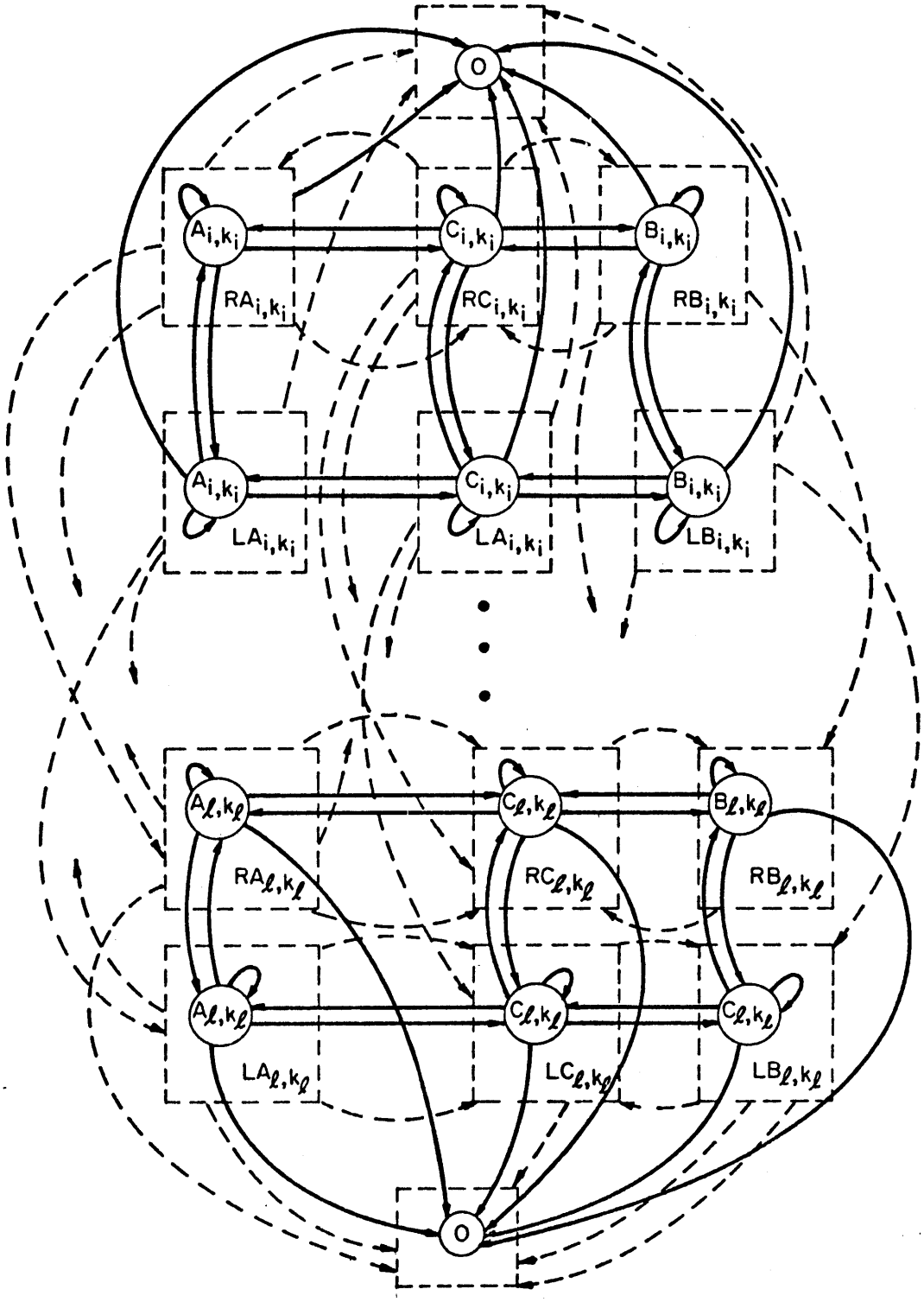
$$\text{Reactants of products region} = \{(z_1,z_2,z_3) \in Z^3 \; z_2 \geq 2, \; z_3 = 0, \; \min \leq z_1 \leq \max\} \qquad (7),$$

as indicated in Figure 12[1]. The numbers min and max are determined by the number of operons[2] in the system. In the allocation of elements, these numbers are the z_1-ordinates of STOP elements which constitute <u>barriers</u> for the back and forth motion of transformation elements along the z_1-ordinate ($z_2 = 2$, $z_3 = 0$), as explained below.

---

[1]As we shall see, biomolecular transformation elements and control metabolite elements are also allocated in the region defined by (7).

[2]The reason for this limitation is purely operational. Its purpose is to keep the reactants in the interesting region of the information space.



State Diagram of the Transformation Element of Bimolecular Type

Figure 14

Table 18

Structure Transition Function of the Transformation Bimolecular Element

Present Structure (Reaction System)	Present Intensity					Next Structure (Reaction System)
	Neighbor 0	Neighbor 1	Neighbor 4	Neighbor 3	Neighbor 5	
RA_i,k_i	A_i,k_i	--	--	--	A_i,k_i	RC_i,k_i
RA_i,k_i	A_i,k_i	--	A_ℓ,k_ℓ	--	0	RA_ℓ,k_ℓ
RA_i,k_i	A_i,k_i	--	B_ℓ,k_ℓ	--	0	RB_ℓ,k_ℓ
RA_i,k_i	A_i,k_i	--	C_ℓ,k_ℓ	--	0	RC_ℓ,k_ℓ
RA_i,k_i	A_i,k_i	any≠stop	--	A_ℓ,k_ℓ	--	RA_ℓ,k_ℓ
RA_i,k_i	A_i,k_i	any≠stop	--	B_ℓ,k_ℓ	--	RB_ℓ,k_ℓ
RA_i,k_i	A_i,k_i	any≠stop	--	C_ℓ,k_ℓ	--	RC_ℓ,k_ℓ
RA_i,k_i	A_i,k_i	stop	--	--	--	LA_i,k_i
LA_i,k_i	A_i,k_i	--	--	--	A_i,k_i	RC_i,k_i
RA_i,k_i	A_i,k_i	--	--	--	AB_i,k_i	0
LA_i,k_i	A_i,k_i	--	--	--	AB_i,k_i	0
LA_i,k_i	A_i,k_i	A_ℓ,k_ℓ	--	any≠stop	--	LA_ℓ,k_ℓ
LA_i,k_i	A_i,k_i	B_ℓ,k_ℓ	--	any≠stop	--	LB_ℓ,k_ℓ
LA_i,k_i	A_i,k_i	C_ℓ,k_ℓ	--	any≠stop	--	LC_ℓ,k_ℓ
LA_i,k_i	A_i,k_i	--	--	stop	--	LA_i,k_i
LA_i,k_i	A_i,k_i	--	A_ℓ,k_ℓ	--	0	LA_ℓ,k_ℓ
LA_i,k_i	A_i,k_i	--	B_ℓ,k_ℓ	--	0	LB_ℓ,k_ℓ
LA_i,k_i	A_i,k_i	--	C_ℓ,k_ℓ	--	0	LC_ℓ,k_ℓ

Table 18 (contd.)

Present Structure (Reaction System)	Present Intensity					Next Structure (Reaction System)
	Neighbor 0	Neighbor 1	Neighbor 4	Neighbor 3	Neighbor 5	
RB_i,k_i	B_i,k_i	--	--	--	B_i,k_i	RC_i,k_i
RB_i,k_i	B_i,k_i	--	A_ℓ,k_ℓ	--	0	RA_ℓ,k_ℓ
RB_i,k_i	B_i,k_i	--	B_ℓ,k_ℓ	--	0	RB_ℓ,k_ℓ
RB_i,k_i	B_i,k_i	--	C_ℓ,k_ℓ	--	0	RC_ℓ,k_ℓ
RB_i,k_i	B_i,k_i	any≠stop	--	A_ℓ,k_ℓ	--	RA_ℓ,k_ℓ
RB_i,k_i	B_i,k_i	any≠stop	--	B_ℓ,k_ℓ	--	RB_ℓ,k_ℓ
RB_i,k_i	B_i,k_i	any≠stop	--	C_ℓ,k_ℓ	--	RC_ℓ,k_ℓ
RB_i,k_i	B_i,k_i	stop	--	--	--	LB_i,k_i
LB_i,k_i	B_i,k_i	--	--	--	B_i,k_i	LC_i,k_i
RB_i,k_i	B_i,k_i	--	--	--	AB_i,k_i	0
LB_i,k_i	B_i,k_i	--	--	--	AB_i,k_i	0
LB_i,k_i	B_i,k_i	A_ℓ,k_ℓ	--	any≠stop	--	LA_ℓ,k_ℓ
LB_i,k_i	B_i,k_i	B_ℓ,k_ℓ	--	any≠stop	--	LB_ℓ,k_ℓ
LB_i,k_i	B_i,k_i	C_ℓ,k_ℓ	--	any≠stop	--	LC_ℓ,k_ℓ
LB_i,k_i	B_i,k_i	--	--	stop	--	LB_i,k_i
LB_i,k_i	B_i,k_i	--	A_ℓ,k_ℓ	--	0	LA_ℓ,k_ℓ
LB_i,k_i	B_i,k_i	--	B_ℓ,k_ℓ	--	0	LB_ℓ,k_ℓ
LB_i,k_i	B_i,k_i	--	C_ℓ,k_ℓ	--	0	LC_ℓ,k_ℓ

Table 18 (contd.)

Present Structure (Reaction System)	Present Intensity					Next Structure (Reaction System)
	Neighbor 0	Neighbor 1	Neighbor 4	Neighbor 3	Neighbor 5	
RC_i,k_i	C_i,k_i	--	--	--	A_i,k_i	(RB_i,k_i)
RC_i,k_i	C_i,k_i	--	--	--	B_i,k_i	(RA_i,k_i)
RC_i,k_i	C_i,k_i	--	--	--	AB_i,k_i	(0)
RC_i,k_i	C_i,k_i	--	A_ℓ,k_ℓ	--	0	RA_ℓ,k_ℓ
RC_i,k_i	C_i,k_i	--	B_ℓ,k_ℓ	--	0	RB_ℓ,k_ℓ
RC_i,k_i	C_i,k_i	--	C_ℓ,k_ℓ	--	0	RC_ℓ,k_ℓ
RC_i,k_i	C_i,k_i	stop	--	--	--	LC_i,k_i
LC_i,k_i	C_i,k_i	--	--	stop	--	RC_i,k_i
LC_i,k_i	C_i,k_i	B_ℓ,k_ℓ	--	any≠stop	--	LB_ℓ,k_ℓ
LC_i,k_i	C_i,k_i	A_ℓ,k_ℓ	--	any≠stop	--	LA_ℓ,k_ℓ
LC_i,k_i	C_i,k_i	C_ℓ,k_ℓ	--	any≠stop	--	LC_ℓ,k_ℓ
RC_i,k_i	C_i,k_i	any≠stop	--	A_ℓ,k_ℓ	--	RA_ℓ,k_ℓ
RC_i,k_i	C_i,k_i	any≠stop	--	B_ℓ,k_ℓ	--	RB_ℓ,k_ℓ
RC_i,k_i	C_i,k_i	any≠stop	--	C_ℓ,k_ℓ	--	RC_ℓ,k_ℓ
LC_i,k_i	C_i,k_i	--	--	--	A_i,k_i	LB_i,k_i
LC_i,k_i	C_i,k_i	--	--	--	B_i,k_i	(LA_i,k_i)
LC_i,k_i	C_i,k_i	--	--	--	AB_i,k_i	(0)
LC_i,k_i	C_i,k_i	--	A_ℓ,k_ℓ	--	0	LA_ℓ,k_ℓ
LC_i,k_i	C_i,k_i	--	B_ℓ,k_ℓ	--	0	LB_ℓ,k_ℓ
LC_i,k_i	C_i,k_i	--	C_ℓ,k_ℓ	--	0	LC_ℓ,k_ℓ

Table 19

Intensity Transition Functions for the Transformation Elements RA_i,k_i

Present Intensity (Reaction System)	Present Intensity				Next Intensity (Reaction System)
	Neighbor 1	Neighbor 4	Neighbor 3	Neighbor 5	
A_i,k_i	--	--	--	A_i,k_i	C_i,k_i
A_i,k_i	--	A_ℓ,k_ℓ	--	0	A_ℓ,k_ℓ
A_i,k_i	--	B_ℓ,k_ℓ	--	0	B_ℓ,k_ℓ
A_i,k_i	--	C_ℓ,k_ℓ	--	0	C_ℓ,k_ℓ
A_i,k_i	stop	--	--	--	A_i,k_i
A_i,k_i	any≠stop	--	A_ℓ,k_ℓ	--	A_ℓ,k_ℓ
A_i,k_i	any≠stop	--	B_ℓ,k_ℓ	--	B_ℓ,k_ℓ
A_i,k_i	any≠stop	--	C_ℓ,k_ℓ	--	C_ℓ,k_ℓ
A_i,k_i	--	--	--	AB_i,k_i	0

LA_i,k_i

Present Intensity (Reaction System)	Present Intensity				Next Intensity (Reaction System)
	Neighbor 3	Neighbor 4	Neighbor 1	Neighbor 5	
A_i,k_i	--	--	--	A_i,k_i	C_i,k_i
A_i,k_i	--	--	--	AB_i,k_i	0
A_i,k_i	--	A_ℓ,k_ℓ	--	0	A_ℓ,k_ℓ
A_i,k_i	--	B_ℓ,k_ℓ	--	0	B_ℓ,k_ℓ
A_i,k_i	--	C_ℓ,k_ℓ	--	0	C_ℓ,k_ℓ
A_i,k_i	stop	--	--	--	A_i,k_i
A_i,k_i	any≠stop	--	A_ℓ,k_ℓ $\ell\neq i$	--	A_ℓ,k_ℓ
A_i,k_i	any≠stop	--	B_ℓ,k_ℓ $\ell\neq i$	--	B_ℓ,k_ℓ
A_i,k_i	any≠stop	--	C_ℓ,k_ℓ $\ell\neq i$	--	C_ℓ,k_ℓ

Table 19 (contd.)

RB_i,k_i

Present Intensity (Reaction System)	Present Intensity				Next Intensity (Reaction System)
	Neighbor 1	Neighbor 4	Neighbor 3	Neighbor 5	
B_i,k_i	--	--	--	B_i,k_i	C_i,k_i
B_i,k_i	--	A_ℓ,k_ℓ	--	0	A_ℓ,k_ℓ
B_i,k_i	--	B_ℓ,k_ℓ	--	0	B_ℓ,k_ℓ
B_i,k_i	--	C_ℓ,k_ℓ	--	0	C_ℓ,k_ℓ
B_i,k_i	stop	--	--	--	B_i,k_i
B_i,k_i	any≠stop	--	A_ℓ,k_ℓ	--	A_ℓ,k_ℓ
B_i,k_i	any≠stop	--	B_ℓ,k_ℓ	--	B_ℓ,k_ℓ
B_i,k_i	any≠stop	--	C_ℓ,k_ℓ	--	C_ℓ,k_ℓ
B_i,k_i	--	--	--	AB_i,k_i	0

LB_i,k_i

Present Intensity (Reaction System)	Present Intensity				Next Intensity (Reaction System)
	Neighbor 3	Neighbor 4	Neighbor 1	Neighbor 5	
B_i,k_i	--	--	--	B_i,k_i	C_i,k_i
B_i,k_i	--	--	--	AB_i,k_i	0
B_i,k_i	--	A_ℓ,k_ℓ	--	0	A_ℓ,k_ℓ
B_i,k_i	--	B_ℓ,k_ℓ	--	0	B_ℓ,k_ℓ
B_i,k_i	--	C_ℓ,k_ℓ	--	0	C_ℓ,k_ℓ
B_i,k_i	stop	--	--	--	B_i,k_i
B_i,k_i	any≠stop	--	A_ℓ,k_ℓ $\ell\neq i$	--	A_ℓ,k_ℓ
B_i,k_i	any≠stop	--	B_ℓ,k_ℓ $\ell\neq i$	--	B_ℓ,k_ℓ
B_i,k_i	any≠stop	--	C_ℓ,k_ℓ $\ell\neq i$	--	C_ℓ,k_ℓ

Table 19 (contd.)

RC_{i},k_{i}

Present Intensity (Reaction System)	Present Intensity				Next Intensity (Reaction System)
	Neighbor 1	Neighbor 4	Neighbor 3	Neighbor 5	
C_{i},k_{i}	--	--	--	A_{i},k_{i}	(B_{i},k_{i})[1]
C_{i},k_{i}	--	--	--	B_{i},k_{i}	(A_{i},k_{i})
C_{i},k_{i}	--	--	--	AB_{i},k_{i}	(0)
C_{i},k_{i}	--	A_{ℓ},k_{ℓ}	--	0	A_{ℓ},k_{ℓ}
C_{i},k_{i}	--	B_{ℓ},k_{ℓ}	--	0	B_{ℓ},k_{ℓ}
C_{i},k_{i}	--	C_{ℓ},k_{ℓ}	--	0	C_{ℓ},k_{ℓ}
C_{i},k_{i}	stop	--	--	--	C_{i},k_{i}
C_{i},k_{i}	any≠stop	--	A_{ℓ},k_{ℓ}	--	A_{ℓ},k_{ℓ}
C_{i},k_{i}	any≠stop	--	B_{ℓ},k_{ℓ}	--	B_{ℓ},k_{ℓ}
C_{i},k_{i}	any≠stop	--	C_{ℓ},k_{ℓ}	--	C_{ℓ},k_{ℓ}

LC_{i},k_{i}

Present Intensity (Reaction System)	Present Intensity				Next Intensity (Reaction System)
	Neighbor 3	Neighbor 4	Neighbor 1	Neighbor 5	
C_{i},k_{i}	--	--	--	A_{i},k_{i}	(B_{i},k_{i})
C_{i},k_{i}	--	--	--	B_{i},k_{i}	(A_{i},k_{i})
C_{i},k_{i}	--	--	--	AB_{i},k_{i}	(0)
C_{i},k_{i}	--	A_{ℓ},k_{ℓ}	--	0	A_{ℓ},k_{ℓ}
C_{i},k_{i}	--	B_{ℓ},k_{ℓ}	--	0	B_{ℓ},k_{ℓ}
C_{i},k_{i}	--	C_{ℓ},k_{ℓ}	--	0	C_{ℓ},k_{ℓ}
C_{i},k_{i}	stop	--	--	--	C_{i},k_{i}
C_{i},k_{i}	any≠stop	--	B_{ℓ},k_{ℓ}	--	B_{ℓ},k_{ℓ}
C_{i},k_{i}	any≠stop	--	A_{ℓ},k_{ℓ}	--	A_{ℓ},k_{ℓ}
C_{i},k_{i}	any≠stop	--	C_{ℓ},k_{ℓ}	--	C_{ℓ},k_{ℓ}

[1] ( ) indicates "exclusively if $A_{i},k_{i} + B_{i},k_{i} \underset{PR_{i},k_{i}}{\overset{\rightarrow}{\leftarrow}} C_{i},k_{i}$ is reversible".

When in state ($R\ell_{i,k_i}$, A_{i,k_i}) or, ($R\ell_{i,k_i}$, B_{i,k_i}) or, ($R\ell_{i,k_i}$, A_{i,k_i}) or, ($R\ell_{i,k_i}$, B_{i,k_i}) if the monomolecular transformation element allocated at some point in the reactants and products region has as its 4 neighbor an element in intensity A_{ℓ,k_ℓ} or B_{ℓ,k_ℓ} the next state of the monomolecular element becomes (Rr_{ℓ,k_ℓ}, A_{ℓ,k_ℓ}) or (Rr_{ℓ,k_ℓ}, B_{ℓ,k_ℓ}), respectively (while that if its 2-nearest neighbor becomes (Rr_{i,k_i}, A_{i,k_i}) or (Rr_{i,k_i}, B_{i,k_i}) or ($R\ell_{i,k_i}$, A_{i,k_i}) or ($R\ell_{i,k_i}$, B_{i,k_i}) thus the element "moves" under this state transition in the negative direction of the z_2-ordinate. This motion continues until the element reaches the ordinate $z_2=2$, which is detected by the fact that its 5-nearest neighbor has a non-dormant intensity (see Figure 12 and Tables 16 and 17).

When the monomolecular element is at some point along $z_2=2$, $z_3=0$ and in state ($R\ell_{i,k_i}$, A_{i,k_i}) (or ($R\ell_{i,k_i}$, A_{i,k_i})) and its 5-nearest neighbor element (a protein element) is in intensity A_{i,k_i}, the element suffers a state transition to state (Rr_{i,k_i}, B_{i,k_i}) (or ($R\ell_{i,k_i}$, B_{i,k_i})) indicating the completion of a monomolecular reversible or irreversible reaction.[1] If the intensity of its 5-nearest neighbor is A_{m,k_m} or B_{m,k_m} (i.e., corresponding to protein PR_{m,k_m} and the intensity of its 3-neighbor (or its 1-neighbor) it A_{ℓ,k_ℓ} or B_{ℓ,k_ℓ} the state of the molecular element becomes (Rr_{ℓ,k_ℓ}, A_{ℓ,k_ℓ}) or (Rr_{ℓ,k_ℓ}, B_{ℓ,k_ℓ}), (or ($R\ell_{\ell,k_\ell}$,

[1]The information of whether the reaction is reversible or irreversible is contained in the state structure of PR_{i,k_i}, as explained previously.

A_{ℓ,k_ℓ}), or R_{ℓ,k_ℓ}, B_{ℓ,k_ℓ})) while that of its 1-neighbor (or its 3-neighbor) becomes (Rr_{i,k_i}, A_{i,k_i}) (or $R\ell_{i,k_i}$, A_{i,k_i}). Thus either the reaction is completed (if the appropriate protein element Pr_{i,k_i} is in the catalysis region) or the element is transported one step in the positive direction of z_i (or the negative direction), or the transformation element finds a STOP element as its 1-nearest neighbor (or its 3-nearest neighbor), as mentioned earlier (see also Tables 16 and 17).

Now we pass to describe the state representation of transformation elements of bimolecular type, for both irreversible and reversible reaction systems. A state diagram for the type of these elements is shown in Figure 14. The corresponding structure and intensity transition functions are given in Tables 18 and 19, respectively.

The state representation of bimolecular transformation elements is composed of 7 classes of structures: $\{RA_{i,k_i}\}$,$\{RB_{i,k_i}\}$, $\{RC_{i,k_i}\}$, $\{LA_{i,k_i}\}$ $\{B_{i,k_i}\}$, $\{LC_{i,k_i}\}$ and the dormant structure. The first 3 structures are labels for the element representing the 2 reactants and the product of a bimolecular reaction system in Right motion mode. The last 3 represent these molecules in Left motion mode. The operational objective of these structures is explained next.

As can be inferred from a comparison of Tables 18 and 19 with Tables 16 and 17 respectively, the motion of bimolecular elements in the reactant and products region and along the z_1-coordinate corresponding to $z_2=2$ $z_3=0$ is analogous to that of monomolecular elements if we identify structures $\{RA_{i,k_i}\}$ or $\{RB_{i,k_i}\}$ or $\{RC_{i,k_i}\}$ with $\{Rr_{i,k_i}\}$ and structures

$\{LA_{i,k_i}\}$ or $\{LB_{i,k_i}\}$ or $\{LC_{i,k_i}\}$ with $\{R\ell_{i,k_i}\}$. Therefore, our discussion will be restricted to the catalytic characteristics of these bimolecular elements.

Reactions of the form

$$A_{i,k_i} + B_{i,k_i} \xrightarrow[PR_{i,k_i}]{} C_{i,k_i}$$

and

$$A_{i,k_i} + B_{i,k_i} \underset{PR_{i,k_i}}{\overset{\rightarrow}{\leftarrow}} C_{i,k_i}$$

As mentioned earlier, when we discussed the protein elements that catalyze these types of reactions, we represent the dynamics of these reactions as a 2 step (not necessarily consecutive step) state transition.

If the bimolecular element is in state (RA_{i,k_i}, A_{i,k_i}) (or (RB_{i,k_i}, B_{i,k_i})) and the element is its 5-nearest neighbor location (a protein; see Figure 12) is in intensity AB_{i,k_i} the element's state becomes (0,0) indicating the disappearance of the system of one A_{i,k_i} molecule (resp. one B_{i,k_i} molecule). The corresponding protein element now represents the <u>complex</u> $A_{i,k_i} - PR_{i,k_i}$ (Resp. $Pr_{i,k_i} - B_{i,k_i}$) the reaction is then completed when this element finds in its 6-nearest neighbor and element in intensity B_{i,k_i}, (Resp. A_{i,k_i}) representing a B_{i,k_i} molecule, at which time the state of the protein element, which now represents the complex, changes to $(RS''_{i,k_i}, C_{i,k_i})$ (or $(RS'''_{i,k_i}, C_{i,k_i})$ if the reaction is reversible) and that of the molecule changes to C_{i,k_i} indicating the

completion of the reaction. We want to point out that this operational behavior (i.e., the formation of complexes of protein and one of the reactants) is generally accepted to be the biochemical mechanism by which at least some bimolecular reactions evolve. It is interesting that this behavior is advantageous since very seldom could we find, simultaneously, in appropriate neighboring locations the two reactants that participate in bimolecular reactions.

A similar operational behavior is observed for the reversible case when the state of the transformation bimolecular element is (RC_{i,k_i}, C_{i,k_i}) and the element allocated at its 5-nearest-neighboring point is state (RS''_{i,k_i}, C_{i,k_i}).

We close our digression for the description of transformation elements with an important observation about the labeling of their intensities. Recall that the element A_{i,k_i} is a molecule which is a reactant in the k_i-th reaction system associated with the i-th operon, that is A is a reactant in a reaction of the form $A_{i,k_i} \xrightarrow[PR_{i,k_i}]{} B_{i,k_i}$, or,

$A_{i,k_i} \underset{PR_{i,k_i}}{\overset{\rightarrow}{\leftarrow}} B_{i,k_i}$, or $A_{i,k_i} + B_{i,k_i} \xrightarrow[PR_{i,k_i}]{} C_{i,k_i}$, or $A_{i,k_i} + B_{i,k_i} \underset{PR_{i,k_i}}{\overset{\rightarrow}{\leftarrow}} C_{i,k_i}$.

This does not imply that this molecule is uniquely associated with the k_i-th reaction system. For instance, we can have the following situation

$$\begin{array}{l} X + Y \xrightarrow[PR_{i,k_i}]{} Z \\ Y_1 \xrightarrow[PR_{j,k_j}]{} X \end{array} \qquad (8)$$

In our notation, this is written as

$$A_{i,k_i} + B_{i,k_i} \xrightarrow[PR_{i,k_i}]{} C_{i,k_i} \qquad (9)$$

$$A_{j,k_j} \xrightarrow[PR_{j,k_j}]{} B_{j,k_j}$$

with $A_{i,k_i} = B_{j,k_j} = X$

$B_{i,k_i} = Y$

$C_{i,k_i} = Z$

$A_{j,k_j} = Y_1$

Thus, in the model, the numbers assigned to intensities A_{i,k_i} and B_{j,k_j} are equal.

Now we discuss the state structure of a fifth type of protein element; the element representing regulator proteins. (See section 2.1, Figure 2.1-5 and companion discussion.)

A state diagram for the type of regulator proteins is given in Figure 10-e. The corresponding structure and intensity transition functions are given in tables 20 and 21, respectively. As in the previous four cases we give the type of regulator proteins as an element capable of regulating one of m different operons[1].

[1]As we shall see in a moment, the action of regulatory proteins is simulated in our model by a particular interaction between them and the corresponding TFM.

Table 20

Structure Transition Function of Protein Elements of Regulator Type

Present Structure (Protein)	Present Intensity							Next Structure (Protein)
	Neighbor 0	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
Ne_4	0	--	--	--	--	--	cont	$select_4$
Ne_4	--	--	--	--	--	--	any≠cont	Ne_4
$select_4$	$j \neq T_i, k_i$	--	--	--	--	--	cont	$select_4$
$select_4$	$j = T_i, k_i$	--	--	--	--	--	Br	RR_i, k_i
$select_4$	--	--	--	--	--	--	Decont	Ne_4
RR_i, k_i	REG_i, k_i	--	--	--	REG_ℓ, k_ℓ	--	--	RR_ℓ, k_ℓ
RR_i, k_i	REG_i, k_i	--	REG_ℓ, k_ℓ	--	--	--	--	RR_ℓ, k_ℓ
RR_i, k_i	REG_i, k_i	--	--	REG	--	--	REG_ℓ, k_ℓ	RR_ℓ, k_ℓ
RR_i, k_i	P_i, k_i	--	--	REG_ℓ, k_ℓ P_ℓ, k_ℓ	--	any≠P_i, k_i	--	RR_ℓ, k_ℓ
RR_i, k_i	P_i, k_i	--	--	--	--	P_i, k_i	--	RR_i, k_i
RR_i, k_i	P_i, k_i	stop	--	--	--	--	--	LR_i, k_i
LR_i, k_i	P_i, k_i	REG_ℓ, k_ℓ P_ℓ, k_ℓ	--	--	--	any≠P_i, k_i	--	LR_ℓ, k_ℓ
LR_i, k_i	P_i, k_i	--	--	--	--	P_i, k_i	--	LR_i, k_i
LR_i, k_i	P_i, k_i	--	--	stop	--	--	--	RR_i, k_i
LR_i, k_i	REG_i, k_i	REG	--	--	--	--	--	LR_i, k_i
RR_i, k_i	De	--	--	--	--	--	stop	$select_4$
LR_i, k_i	De	--	--	--	--	--	stop	$select_4$

Table 21

Intensity Transition Functions of Protein Elements of Regulator Type

Ne_4

Present Intensity (Protein)	Present Intensity (Ribosome)	Next Intensity (Protein)
0	any $\neq$ cont	0
0	cont	1

$select_4$

Present Intensity (Protein)	Present Intensity (Ribosome)	Next Intensity (Protein)
$j \neq T_i,k_i \quad \forall\ i$	cont	j+1
$j = T_i,k_i$	Br	REG_i,k_i
j	Decont	0

RR_{i},k_{i}

Table 21 (contd.)

Present Intensity (Protein)	Present Intensity						Next Intensity (Protein)
	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
REG_{i},k_{i}	--	--	--	REG_{ℓ},k_{ℓ}	$\neq P_{i},k_{i}$	--	REG_{ℓ},k_{ℓ}
REG_{i},k_{i}	--	REG_{ℓ},k_{ℓ}	--	--	--	--	REG_{ℓ},k_{ℓ}
REG_{i},k_{i}	--	--	REG	--	--	--	P_{i},k_{i}
P_{i},k_{i}	--	--	--	REG_{ℓ},k_{ℓ} P_{ℓ},k_{ℓ}	$\neq P_{i},k_{i}$	--	REG_{ℓ},k_{ℓ} P_{ℓ},k_{ℓ}
REG_{i},k_{i}	--	--	--	--	--	REG_{ℓ},k_{ℓ}	REG_{ℓ},k_{ℓ}
REG_{i},k_{i}	stop	--	--	--	--	--	REG_{i},k_{i}
P_{i},k_{i}	stop	--	--	--	--	--	P_{i},k_{i}
P_{i},k_{i}	--	--	P_{ℓ},k_{ℓ}	--	$\neq P_{i},k_{i}$	--	P_{ℓ},k_{ℓ}
REG_{i},k_{i}	--	--	REG_{ℓ},k_{ℓ}	--	$\neq P_{i},k_{i}$	--	REG_{ℓ},k_{ℓ}
P_{i},k_{i}	--	--	--	--	P_{i},k_{i}	--	De
De	--	--	--	--	--	--	0

Table 21 (contd.)

LR_{i},k_{i}

Present Intensity (Protein)	Present Intensity						Next Intensity (Protein)
	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
REG_{i},k_{i}	REG_{ℓ},k_{ℓ} or P_{ℓ},k_{ℓ}	--	--	--	$\neq P_{i},k_{i}$	--	REG_{ℓ},k_{ℓ} or P_{ℓ},k_{ℓ}
REG_{i},k_{i}	REG	--	--	--	--	--	P_{i},k_{i}
P_{i},k_{i}	REG_{ℓ},k_{ℓ} or P_{ℓ},k_{ℓ}	--	--	--	$\neq P_{i},k_{i}$	--	REG_{ℓ},k_{ℓ} or P_{ℓ},k_{ℓ}
P_{i},k_{i}	--	--	--	--	P_{i},k_{i}	--	De
De	--	--	--	--	--	--	0
REG_{i},k_{i}	--	--	--	--	--	REG_{ℓ},k_{ℓ}	REG_{ℓ},k_{ℓ}
REG_{i},k_{i}	--	--	stop	--	--	--	REG_{i},k_{i}
P_{i},k_{i}	--	--	stop	--	--	--	P_{i},k_{i}
REG_{i},k_{i}	--	--	--	REG_{ℓ},k_{ℓ} or P_{ℓ},k_{ℓ}	$\neq P_{i},k_{i}$	--	REG_{ℓ},k_{ℓ} or P_{ℓ},k_{ℓ}
REG_{i},k_{i}	--	REG_{ℓ},k_{ℓ}	--	--	$\neq P_{i},k_{i}$	--	REG_{ℓ},k_{ℓ}

The state structure of a regulator protein element is composed of 4 classes of structures: Ne_4, $select_4$, $\{RR_{i,k_i}\}$ and $\{LR_{i,k_i}\}$. The operational objectives of Ne_4 and $select_4$ are analogous to those of Ne and select in protein elements of monomolecular irreversible type, (see Tables 12 and 13 and companion discussion) which are, to simulate the synthesis (by the ribosome) of the regulator protein (the simulation is carried out by $select_4$ which "counts" up to the appropriate time-of-synthesis of the molecule).

The operational objective of structure RR_{i,k_i} is to transport the molecule from the catalysis to the regulation regions (see Figure 12) which is the set of points in the informational space defined by

$$\text{Regulation Region} = \{(z_1,z_2,z_3) \in Z^3 \mid z_2=1, z_3=1, \min \leq z_1 \leq \max\} \tag{10}$$

where min and max were defined earlier (see expression 7), and to search for the appropriated transcription feedback mechanism (TFM) in its 5-nearest neighbor location that is regulated by the protein represented by the element. In RR_{i,k_i} the regulator protein PR_{i,k_i} is moving along the regulation region in the positive direction of the z_1-axis until either the protein finds the appropriate transcription feedback mechanism (in as will be explained below) or it finds a virtual element (in z_1=max) STOP in its 1-nearest neighboring location. In the second situation the structure of the element changes to LR_{i,k_i} in which the protein element moves along the regulation region in the negative direction of the

z_1-axis until the protein finds the appropriate transcription feedback mechanism or it finds a STOP element in its 3-nearest neighboring location in which case the structure changes to RR_{i,k_i}.

When in state (RR_{i,k_i}, REG_{i,k_i}) the regulator protein, at a point with coordinates $z_2=1$, $z_3=1$ and some $z_1 \leq \max$, and with the intensity of its nearest 1-neighbor element being REG[1] (see Tables 20 and 21 and Figure 12) the protein element state changes to (RR_{i,k_i}, P_{i,k_i}). In this state, the protein element is ready to perform its regulation role; which is accomplished if its 5-nearest element (a TFM whose state structure is described later) is in intensity REG_{ℓ,k_ℓ} or P_{ℓ,k_ℓ} its state becomes (RR_{ℓ,k_ℓ}, REG_{ℓ,k_ℓ}) or (RR_{ℓ,k_ℓ}, P_{ℓ,k_ℓ}) respectively, while that of its 1-nearest neighbor becomes (RR_{i,k_i}, P_{i,k_i}). Thus, the element has effected a translation of one point in the positive direction of the z_1-axis. If the intensity of its 5-nearest neighbor is P_{i,k_i} the state of the element becomes (RR_{i,k_i}, De) which represents the protein <u>bound</u> to the TFM regulated by it.

The transport of the protein PR_{i,k_i} element in the negative direction of the z_1-axis (structure (R_{i,k_i})) in the regulation region is accomplished by the symmetrical set of state transitions of the ones described in the last paragraph (see Table 20).

Now we describe the state structure of the last type of protein element considered in our model: the cycle protein (see Section 2.1).

---

[1]REG is a virtual element whose only state is (REG,REG).

A state diagram for these protein types is given in Figure 10-f; the corresponding structure and intensity functions are given in Tables 22 and 23 respectively.

The state diagram for protein elements of cycle type shown in Figure 10-f was constructed by modeling each of the four reaction systems composing the cycle (see Figure 2.1-4) as the reversible step in 4 bimolecular reversible protein elements a prototype of which was discussed earlier. In Chapter 5 we will illustrate with an example its cyclic behavior.

We conclude our discussion of the state structure of protein elements with a state structure which includes the six types described above. Towards this objective, we note two similarities among these six types:

1 -- The inactive status for all the types prior to synthesis ((Ne, Ne_i i=1,...,5), 0) and the synthesis status ((select$_i$, i=1,...,5), j) are identical for all the six types.

2 -- The Motion simulation (via state transitions) from the sunthesis to either the catalysis area or the regulation area and the motion of the elements along these areas is identical (in the sense that corresponding intensities and structures have corresponding equivalent state transitions).

These two observations suggest the following implementation for a type whose state structure includes the types of all six types of proteins (or all the types required in the study of a particular system

Table 22

Structure Transition Function for Protein Element of Cycle Type

Present Structure (Protein)	Present Intensity							Next Structure (Protein)
	Neighbor 0	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
Ne_5	0	--	--	--	--	cont	--	$select_5$
Ne_5	0	--	--	--	--	any≠cont	--	Ne_5
$select_5$	$j \geq 1$ $j \neq T_i, k_i$	--	--	--	--	cont	--	$select_5$
$select_5$	$j = T_i, k_i$	--	--	--	--	Br	--	RA_i, k_i
$select_5$	j	--	--	--	--	Decont	--	Ne_5
RA_i, k_i	A_i, k_i	--	--	--	$k_i \neq k'_i$ A_i, k_i	Br	--	0
RA_i, k_i	A_i, k_i	--	A_i, k'_i	--	$0, A'_i, k''_i$	--	--	RA_i, k'_i
RA_i, k_i	A_i, k_i	--	--	REG	--	A_i, k'_i	--	RA_i, k'_i
RA_i, k_i	A_i, k_i	--	A_ℓ, k_ℓ	--	--	--	--	RA_ℓ, k_ℓ
RA_i, k_i	A_i, k_i	--	--	A_ℓ, k_ℓ	--	--	--	RA_ℓ, k_ℓ
RA_i, k_i	AD_i, k_i A_i, k_i	stop	--	--	--	--	--	RA_i, k_i
RA_i, k_i	A_i, k_i	--	--	--	--	A_i, k_i	--	RA_i, k_i
RA_i, k_i	BD_i, k_i	--	--	--	--	B_i, k_i	--	RD_i, k_i
RA_i, k_i	BD_i, k_i	--	--	--	--	D_i, k_i	--	RB_i, k_i

Table 22 (contd.)

Present Structure (Protein)	Present Intensity							Next Structure (Protein)
	Neighbor 0	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
RB_i,k_i	B_i,k_i	--	--	--	--	B_i,k_i	--	RB_i,k_i
RB_i,k_i	CA_i,k_i	--	--	--	--	C_i,k_i	--	RA_i,k_i
RB_i,k_i	CA_i,k_i	--	--	--	--	A_i,k_i	--	RC_i,k_i
RC_i,k_i	C_i,k_i	--	--	--	--	C_i,k_i	--	RC_i,k_i
RC_i,k_i	AD_i,k_i	--	--	--	--	D_i,k_i	--	RA_i,k_i
RC_i,k_i	AD_i,k_i	--	--	--	--	A_i,k_i	--	RD_i,k_i
RD_i,k_i	D_i,k_i	--	--	--	--	D_i,k_i	--	RD_i,k_i
RD_i,k_i	BC_i,k_i	--	--	--	--	B_i,k_i	--	RC_i,k_i
RD_i,k_i	BC_i,k_i	--	--	--	--	C_i,k_i	--	RB_i,k_i
RA_i,k_i	A_i,k_i	--	--	--	--	--	stop	T_i,k_i
RB_i,k_i	B_i,k_i	--	--	--	--	--	stop	0
RC_i,k_i	C_i,k_i	--	--	--	--	--	stop	0
RD_i,k_i	D_i,k_i	--	--	--	--	--	stop	0

Table 23

Ne_5

Present Intensity (Protein)	Present Intensity (Ribosome)	Next Intensity (Protein)
0	cont	1
0	any≠cont	0

$select_5$

Present Intensity (Protein)	Present Intensity (Ribosome)	Next Intensity (Protein)
$j \geq 1$ $j \neq T_i, k_i$	cont	j+1
$j = T_i, k_i$	Br	A_i, k_i
j	Decont	0

Table 23 (contd.)

RA_{i},k_{i}

Present Intensity (Protein)	Present Intensity						Next Intensity (Protein)
	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
A_{i},k_{i}	--	--	--	A_{i},k'_{i}	--	--	A_{i},k'_{i}
A_{i},k_{i}	--	A_{i},k'_{i}	--	--	--	--	A_{i},k'_{i}
A_{i},k_{i}	--	--	A_{ℓ},k_{ℓ}	--	any$\neq A_{i},k_{i}$	--	A_{ℓ},k_{ℓ}
A_{i},k_{i}	--	--	REG	--	A_{i},k'_{i}	--	A_{i},k'_{i}
A_{i},k_{i}	stop	--	--	--	--	--	A_{i},k_{i}
A_{i},k_{i}	--	--	--	--	A_{i},k_{i}	--	BD_{i},k_{i}
BD_{i},k_{i}	--	--	--	--	B_{i},k_{i}	--	D_{i},k_{i}
A_{i},k_{i}	--	--	--	--	--	stop	T_{i},k_{i}
BD_{i},k_{i}	--	--	--	--	D_{i},k_{i} any$\neq B_{i},k_{i}$	--	BD_{i},k_{i}
BD_{i},k_{i}	--	--	--	--	D_{i},k_{i}	--	B_{i},k_{i}

Table 23 (contd.)

RB_{i},k_{i}

Present Intensity (Protein)	Present Intensity						Next Intensity (Protein)
	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
B_{i},k_{i}	--	--	--	--	--	stop	0
B_{i},k_{i}	--	--	--	--	B_{i},k_{i}	--	CA_{i},k_{i}
B_{i},k_{i}	--	--	--	--	any$\neq B_{i},k_{i}$	--	B_{i},k_{i}
B_{i},k_{i}	--	--	A_{ℓ},k_{ℓ} C_{ℓ},k_{ℓ} B_{ℓ},k_{ℓ} D_{ℓ},k_{ℓ}	--	any$\neq B_{i},k_{i}$	--	A_{ℓ},k_{ℓ} C_{ℓ},k_{ℓ} B_{ℓ},k_{ℓ} D_{ℓ},k_{ℓ}
CA_{i},k_{i}	--	--	--	--	C_{i},k_{i}	--	A_{i},k_{i}
CA_{i},k_{i}	--	--	--	--	A_{i},k_{i}	--	C_{i},k_{i}
CA_{i},k_{i}	--	--	A_{ℓ},k_{ℓ} C_{ℓ},k_{ℓ} B_{ℓ},k_{ℓ} D_{ℓ},k_{ℓ}	--	any$\neq$ C_{i},k_{i} A_{i},k_{i}	--	A_{ℓ},k_{ℓ} C_{ℓ},k_{ℓ} B_{ℓ},k_{ℓ} D_{ℓ},k_{ℓ}
CA_{i},k_{i}	stop	--	--	--	--	--	CA_{i},k_{i}
B_{i},k_{i}	stop	--	--	--	--	--	B_{i},k_{i}

Table 23 (contd.)

Present Intensity (Protein)	Present Intensity						Next Intensity (Protein)
	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
C_i,k_i	--	--	--	--	--	stop	0
C_i,k_i	--	--	--	--	C_i,k_i	--	AD_i,k_i
C_i,k_i	--	--	--	--	any$\neq C_i,k_i$	--	C_i,k_i
C_i,k_i	--	--	A_ℓ,k_ℓ C_ℓ,k_ℓ B_ℓ,k_ℓ D_ℓ,k_ℓ	--	any$\neq C_i,k_i$	--	A_ℓ,k_ℓ C_ℓ,k_ℓ B_ℓ,k_ℓ D_ℓ,k_ℓ
AD_i,k_i	--	--	--	--	D_i,k_i	--	A_i,k_i
AD_i,k_i	--	--	--	--	A_i,k_i	--	D_i,k_i
AD_i,k_i	--	--	A_ℓ,k_ℓ C_ℓ,k_ℓ B_ℓ,k_ℓ D_ℓ,k_ℓ	--	any$\neq$ C_i,k_i D_i,k_i	--	A_ℓ,k_ℓ C_ℓ,k_ℓ B_ℓ,k_ℓ D_ℓ,k_ℓ
AD_i,k_i	stop	--	--	--	--	--	AD_i,k_i
C_i,k_i	stop	--	--	--	--	--	C_i,k_i

Table 23 (contd.)

RD_{i},k_{i}

Present Intensity (Protein)	Present Intensity						Next Intensity (Protein)
	Neighbor 1	Neighbor 2	Neighbor 3	Neighbor 4	Neighbor 5	Neighbor 6	
D_{i},k_{i}	--	--	--	--	--	stop	0
D_{i},k_{i}	--	--	--	--	D_{i},k_{i}	--	BC_{i},k_{i}
D_{i},k_{i}	--	--	--	--	any$\neq D_{i},k_{i}$	--	C_{i},k_{i}
D_{i},k_{i}	--	--	A_{ℓ},k_{ℓ} C_{ℓ},k_{ℓ} B_{ℓ},k_{ℓ} D_{ℓ},k_{ℓ}	--	any$\neq D_{i},k_{i}$	--	A_{ℓ},k_{ℓ} C_{ℓ},k_{ℓ} B_{ℓ},k_{ℓ} D_{ℓ},k_{ℓ}
BC_{i},k_{i}	--	--	--	--	B_{i},k_{i}	--	C_{i},k_{i}
BC_{i},k_{i}	--	--	--	--	C_{i},k_{i}	--	B_{i},k_{i}
BC_{i},k_{i}	--	--	$A_{\ell},k_{\ell}C_{\ell},k_{\ell}$ $B_{\ell},k_{\ell}D_{\ell},k_{\ell}$	--	any$\neq$ B_{i},k_{i} C_{i},k_{i}	--	A_{ℓ},k_{ℓ} C_{ℓ},k_{ℓ} B_{ℓ},k_{ℓ} D_{ℓ},k_{ℓ}
BC_{i},k_{i}	stop	--	--	--	--	--	BC_{i},k_{i}
D_{i},k_{i}	stop	--	--	--	--	--	D_{i},k_{i}

under study since it may happen that one or more of the protein types discussed above is not present in that system).

-- The Inactive structure is Ne and its only intensity is 0.

-- The counter structure is select with a set of intensities $\{T_{i,k_i}\}$, representing the times of synthesis for each protein present in the system under study.

-- Each protein A_{i,k_i} present in the system is assumed to be of one of the six types discussed earlier and consequently, we incorporate to the type one and only one of the following structures RS_{i,k_i}, RS'_{i,k_i}, RS''_{i,k_i}, RS'''_{i,k_i} if the protein catalyzes a monomolecular irreversible or reversible, bimolecular irreversible or reversible reaction, two structures RR_{i,k_i} and LR_{i,k_i} if Pr_{i,k_i} is of regulator type, or 4 structures RA_{i,k_i}, RB_{i,k_i}, RC_{i,k_i}, RD_{i,k_i} is it is of cycle type.

-- It is assumed that <u>all</u> the proteins in the system under study have different times of synthesis.[1]

The state diagram of this "general" protein type is illustrated (schematically) in Figure 11.

Now we conclude our description of the incidence diagram of Figure 1 with a discussion of the state structure of the transcription feedback mechanism type.

---

[1] In the next section we will discuss means to eliminate the ambiquity resulting from the case in which two proteins in the system under study have equal times of synthesis.

A diagram of the state transition structure of TFM is given in Figure 15. The corresponding structure and intensity transition functions are given in Tables 24 and 25 respectively.

We note from Figure 15, that the state structure of TFM is composed of four, mutually independent, substructures. A specific TFM_i (i.e., the one controlling OP_i) will have one and only one of these structures depending on the type of action the corresponding regulator protein exercises (induction or repression) and the type of action of the controlling metabolite (activation or repression) as explained in Section 2.1 (see Figure 2.1-4).

The transcription feedback mechanisms, as indicated in Figure 12, are allocated at the points in the informational space along the ordinate $z_2=1$, $z_3=0$ and some z_1[1]. TFM_i interacts with its 4-neighbor (the control metabolite represented by a transformation element) and its 5-neighbor (the regulator protein represented by a type of the fifth class of protein types described earlier in this section).

The state structure of the transcription feedback mechanism consists of three types of structures $\{CP_i^j\}$, $\{CM_i^j\}$, $\{S_i^j\}$ j=1,2,3,4 where the superindex denotes one and only one of the four possible transcriptional control arrangements described in Section 2.1. We will describe in detail the state transition structure corresponding to a TFM_i for the inducible-

---

[1]The ordinate z_1 of TFM_i coincides with that of OP_i (see Figure 12).

Table 24

Structure Transition Function of Transcription Feedback Mechanism-Element

Present Structure (TFM_i)	Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
CP_i^1	P_i	P_i*	--	CP_i^1
CP_i^1	0	any$\neq P_i$	M_i**	CM_i^1
CP_i^1	0	P_i	any	CP_i^1
CM_i^1	M_i	any$\neq P_i$	M_i	S^1
CM_i^1	M_i	P_i	M_i	CM_i^1
S^1	A	any$\neq P_i$	M_i	S^1
S^1	A	P_i	M_i	S^1
S^1	INA	P_i	any$\neq M_i$	CP_i^1
CP_i^2	0	P_i	any$\neq M_i$	CP_i^2
CP_i^2	P_i	P_i	any$\neq M_i$	CP_i^2
CP_i^2	P_i	any	M_i	CM_i^2
CM_i^2	M_i	any	any$\neq M_i$	CM_i^2
CM_i^2	M_i	P_i	M_i	S^2
S^2	A	P_i	M_i	S^2
S^2	A	any$\neq P_i$	M_i	S^2
S^2	A	P_i	any$\neq M_i$	S^2
S^2	INA	any$\neq P_i$	any$\neq M_i$	CP_i^2
S^2	INA	any$\neq P_i$	--	CP_i^2

*$P_i = P_\ell, k_\ell$ for some ℓ, k_ℓ where P_ℓ, k_ℓ is the intensity of a regulator protein in structure RR ,k or LR ,k (see Tables 20 and 21).

**$M_i = A_s, k_s$ or B_s, k_s or C_s, k_s or D_s, k_s for some s, k_s where these are intensities of the (s, k_s) transformation element.

Table 24 (contd.)

Present Structure (TFM_i)	Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
S^2	INA	any$\neq P_i$	any	CP_i^2
CP_i^3	0	P_i	--	CP_i^3
CP_i^3	P_i	any$\neq P_i$	--	CP_i^3
CP_i^3	0	any$\neq P_i$	any$\neq M_i$	S^3
S^3	A	any$\neq P_i$	any$\neq M_i$	S^3
S^3	A	any	any$\neq M_i$	S^3
S^3	INA	P_i	any$\neq M_i$	S^3
S^3	INA	any	M_i	CM_i^3
CM_i^3	M_i	any$\neq P_i$	M_i	CM_i^3
CM_i^3	M_i	P_i	any	CP_i^3
CP_i^4	P_i	P_i	any$\neq M_i$	S^4
CP_i^4	0	any$\neq P_i$	any	CP_i^4
S^4	A	P_i	any$\neq M_i$	S^4
S^4	INA	P_i	any$\neq M_i$	S^4
S^4	A	P_i	M_i	CM_i^4
CM_i^4	M_i	any	M_i	CP_i^4
CM_i^4	M_i	any	any$\neq M_i$	CM_i^4

Table 25

Intensity Transition Functions for TFM Type

CP_i^1

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
P_i	P_i	--	0
0	any$\neq P_i$	M_i	M_i
0	P_i	any	0
P_i	any$\neq P_i$	--	P_i

CM_i^1

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
M_i	any$\neq P_i$	M_i	A
M_i	P_i	M_i	M_i
M_i	P_i	any$\neq M_i$	M_i

S^1

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
A	any$\neq P_i$	M_i	A
A	P_i	M_i	INA
INA	P_i	any$\neq M_i$	P_i
INA	any$\neq P_i$	M_i	A

Table 25 (contd.)

CP_i^2

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
0	P_i	any$\neq M_i$	P_i
P_i	P_i	any$\neq M_i$	P_i
P_i	P_i	M_i	M_i

CM_i^2

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
M_i	--	M_i	A
M_i	--	any$\neq M_i$	M_i

S^2

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
A	P_i	M_i	A
A	any$\neq P_i$	M_i	INA
A	P_i	any$\neq M_i$	INA
INA	any$\neq P_i$	--	0

Table 25 (contd.)

CP_i^3

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
0	P_i	--	P_i
P_i	any$\neq P_i$	--	0
0	any$\neq P_i$	any$\neq M_i$	A
P_i	any$\neq P_i$	M_i	M_i

CM_i^3

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
M_i	any$\neq P_i$	M_i	M_i
M_i	--	any$\neq M_i$	INA

S^3

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
A	any$\neq P_i$	any$\neq M_i$	A
A	P_i	any$\neq M_i$	INA
A	any	M_i	INA
INA	P_i	any	INA
INA	any	M_i	M_i

Table 25 (contd.)

CP_i^4

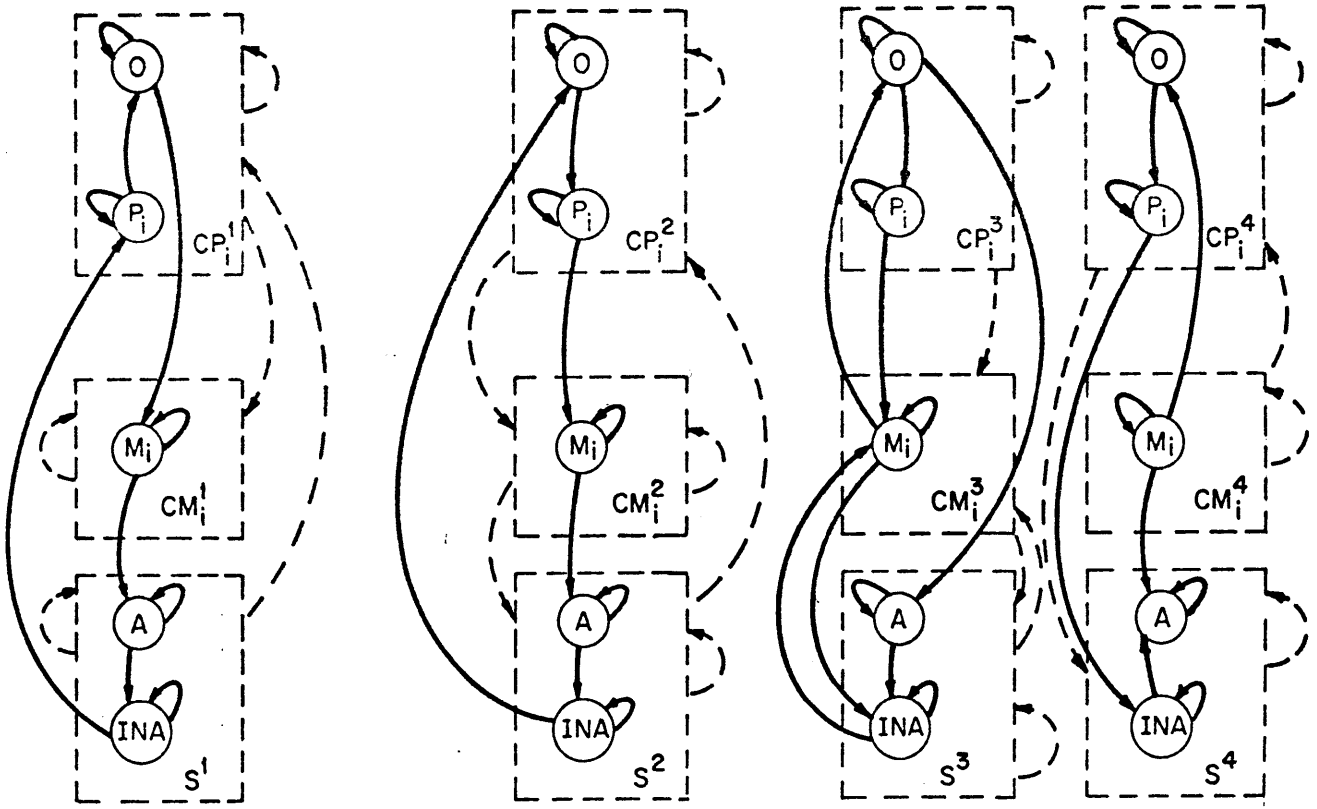
Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
P_i	P_i	any$\neq M_i$	INA
0	any$\neq P_i$	any	0
P_i	P_i	M_i	P_i

CM_i^4

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
M_i	any	M_i	0
M_i	any	any$\neq M_i$	M_i

S^4

Present Intensity (TFM_i)	Present Intensity (Protein)	Present Intensity (Metabolite)	Next Intensity (TFM_i)
A	P_i	any$\neq M_i$	A
INA	P_i	any$\neq M_i$	A
A	P_i	M_i	M_i



Inducible-repressive (See Fig. 2-1.5a)

Inducible-activator (See Fig. 2-1.5b)

Repressible-repressor (See Fig. 2-1.5c)

Repressible-activator (See Fig. 2.1-5d)

$P_i = P_m, k_m$ some m, k_m in one of the proteins of system

$M_i = A_\ell, k_\ell\ \ell, k_\ell$ in one of transformation elements of system

State Diagram of the Transcription-Feedback Mechanism Element (TFM)

Figure 15

repressive arrangement (j=1) the other 3 possibilities have similar state transition substructures and its characteristics can be inferred from the diagram of Figure 15 and from Tables 24 and 25.

When the TFM_i is in structure CP_i' and intensity P_i, the TFM_i (and consequently the corresponding operon OP_i) is repressed (OFF). The state (CP'_{i,P_i}) remains unchanged indicating a previous interaction with the corresponding regulator protein. If the intensity of the element allocated in its 5-nearest neighboring location (a protein) is P_i ($P_i = P_{\ell,k_\ell}$ for some ℓ,k_ℓ) the TFM_i element suffers a state transition to (CP_i', 0) indicating active repression, i.e., the actual presence of the regulator protein. This state remains unchanged until the intensity of the corresponding regulator protein element changes and the intensity of the element allocated in its 4-nearest neighboring location becomes M_i ($M_i = A_{m,k_m}$ for some m,k_m) indicating the absence of repression due to the regulator protein and the presence of the inducing metabolite. Under these conditions the state of TFM_i changes from (CP_i', 0) to (CM'_{i,M_i}).

When in state (CM'_{i,M_i}) the TFM_i element has been induced by the control metabolite. If the intensity of the element representing it (allocated in its 4-nearest neighboring location remains unchanged (i.e., M_i), the state of TFM_i changes to (S',A) indicating that the TFM_i (and consequently OP_i) is in active status (ON, has been discussed before). This state remains unchanged until wither the control metabolite changes its intensity ($\neq M_i$) and/or the intensity of the regulator protein be-

comes P_i. In such cases, the state of TFM_i becomes (S', INA) indicating that TFM_i is inactive. If the state of the regulating protein does not change, the state of TFM_i becomes (CP'_{i,P_i}) and the cycle described in the last paragraph is carried out again.

It is important to note that whenever the state of TFM_i becomes (S', A) at least one copy of OP_i is transcribed as can be inferred from Table 1.

3.4 State Aggregation of the Base Model

In Section 3.1, we established formally that, from an operational point of view, a representation of the dynamic evolution of epigenetic control processes can be obtained by specifying the state transition functions of the participating elements. Some considerations[1] about the microkinetic nature of the molecules (or actions) in epigenetic processes led us to postulate a representation of the operational state of these molecules at any time as an object composed of two items: structure and intensity; with the consequence that the representation of their state transition functions is given by a set of corresponding structure and intensity transition functions.

In Section 3.2 we classified the different molecules and actions that participate in epigenetic control processes into operational (dis-

[1]For instance, high specificity and local (mearest-neighbor) energetic interaction among these molecules or actions.

joint) classes.

Finally, in Section 3.3, we developed a formal state model for a typical element (type) of each of the operational classes established in Section 3.2, under the general criterion adopted in Section 3.1, the accepted dynamic organization and information flow of epigenetic control processes in procaryotes discovered by Jacob and Monod [21], and the fundamental dogma of cellular biology (see Section 2.1). Complementary to the development of the state structure of the types we produced an allocation of the types to specific regions of a 3-dimensional informational space (which simulates the intracellular space) and discussed their local interaction and motions during the state evolution in a simulation of a given epigenetic process.

In this section, we discuss the effects of several simplifying assumptions implicit in our development of Section 3.3 and give arguments to justify them on the basis of the characteristics of the available experimental data (see Section 3.1) with respect to which the quality of the model in predicting epigenetic behavior is to be judged. These assumptions fall into the general category of state aggregation of dynamical systems (see Ziegler [16] and Aoki [1]) and in particular into the subclass known as homomorphic state aggregation, which we will define next in the specific context of our model.

Recall from Sections 3.1 and 3.3 that the state structure of each type τ in the model is given by a structure transition function

$$H_\tau : G_\tau \times (\bigcup_{\gamma \in G_\tau} \mathcal{D}_o f_\tau^\gamma) \to G_\tau \qquad (1)$$

and a finite non-empty set Φ_τ of structure transition functions

$$\Phi_\tau = \{f_\tau^\gamma | \gamma \in G_\tau,\ f_\tau^\gamma : \mathcal{D}_o f_\tau^\gamma \to Q_\tau\} \qquad (2)$$

where Q_τ is the set of intensities of the type, G_τ is the set of labels of elements of Φ_τ (G_τ and Φ_τ are in 1-1 correspondence).

The sets $\mathcal{D}_o f_\tau^\gamma, \gamma \in G_\tau$, are defined as follows

$$\mathcal{D}_o f_\tau^\gamma = \{(q_1,\ldots,q_{n_\gamma}) | q_i \in Q_\tau \ i=1,\ldots,n_\gamma\} \qquad (3)$$

where n_γ = number of arguments (inputs) of the intensity transition function f_τ^γ ($n_\gamma \leq 7 \ \forall \gamma \in G_\tau$ in the types of the model described in Section 3.3).

Clearly, under (2) and (3) $\langle Q_\tau, \Phi_\tau \rangle$ is a universal algebra over Q_τ with operation set Φ_τ (see Appendix A.1). We call this algebra, the <u>canonical algebra of the type τ</u>. This algebra, and some of its properties will be discussed in some detail in Section 4.4 in the construction of a covering type for a given set of types. In here, we use this structure to introduce the concept of <u>intensity aggregation</u> of types.

We say that type τ' (with structure transition function $H_{\tau'}$, intensity transition set $\Phi_{\tau'}$, and intensity set $Q_{\tau'}$) is an <u>intensity-homomorphic aggregation</u> of type τ if the algebra of the former is

(i) Similar[1] to that of the latter

(ii) There exists an epimorphism

$$\phi_1 : \langle Q_\tau, \Phi_\tau \rangle \to \langle Q_{\tau'}, \Phi_{\tau'} \rangle^2$$

such that for every $\gamma \in G_\tau$, $q_1,\ldots,q_{n_\gamma} \in Q$

$$(q_1,\ldots,q_{n_\gamma})f^\gamma\phi_1 = (q_1\phi_1,\ldots,q_{n_\gamma}\phi_1)f^\gamma \qquad (4)$$

By theorem A.1-1 $\langle Q_{\tau'}, \Phi_{\tau'} \rangle$ is isomorphic to the quotient algebra $\langle Q_\tau, \Phi_\tau \rangle / k_\tau$ with k_τ; the kernel of ϕ_1, a congruence on Q_τ. Therefore, for our purposes here, we can identify $\langle Q_{\tau'}, \Phi_{\tau'} \rangle$ with $\langle Q_\tau, \Phi_\tau \rangle / k_\tau$ and define H_{τ/k_τ}, the corresponding structure transition function, and Φ_{τ/k_τ}, the intensity transition set as follows.

Let $\psi_1 : Q_\tau \to Q/k_\tau$ be the map defined by: for $q' \in Q_{\tau'}$, $\psi_1 : q' \to q'\phi_1^{-1}$, (this map is 1-1 by theorem A.1-1). For any $q'_1,\ldots,$

---

[1] See Appendix A.1.

[2] By property (i), we have $\Phi_\tau = \Phi_{\tau'}$. We note that in the model developed in the preceeding section, the various types interact with neighboring elements of different types. In this case, to be mathematically precise, we would have to find the universal type which simulates all of these types (see Section 4.4) and then perform the homomorphic aggregation on this. What we have actually done is to aggregate each type individually and then constructed the universal type for these simpler types. The equivalence of these operations should be clear, and hence we will not dwell on the mathematical inconsistencies between homomorphic aggregation and the actual method of aggregation used.

$q'_{n_{\gamma'}} \in Q_{\tau'}$, $\gamma' \in G_{\tau'} = G_\tau$

$$((q'_1 \psi_1,\ldots,q'_{n_{\gamma'}} \psi_1), \gamma') H_{\tau/k_\tau} = ((q'_1,\ldots,q'_{n_{\gamma'}}),\gamma')H_{\tau'} \quad (5)$$

$$(q'_1 \psi_1,\ldots,q'_{n_{\gamma'}} 1)f^{\gamma'}_{\tau/k_\tau} = (q'_1,\ldots,q'_{n_{\gamma'}})f^{\gamma'}_{\tau'} \psi_1{}^1$$

H_{τ/k_τ} is well defined because ψ_1 is 1-1.

Now we spell out explicitly the instance in which this type of state aggregation (i.e., intensity aggregation) was used in the formulation of the epigenetic model in Section 3.3.

It is well known (see, for example, Watson [ ]) that the genetic code is composed of 64-trinucleotide words (codons), 61 of them coding for one of 20 amino acid residues (not in a 1-1 fashion) one indicating the initial point of a gene, and two for indicating the terminal end of a gene. Thus, from the information-operational point of view, we should have the element representing an operon with two classes of structures.

The first represents initial or terminal status in the transcription of a gene,and the second represents the information content of the genes forming the operon (one for each gene).

In the model of Section 3.3 the two classes of structures mentioned in the last paragraph are labelled respectively,

C,for initiation or termination status

---

[1] ψ_1 is a homomorphism (see theorem A-1.1)

$G_{i,j}$ $j=1,\ldots,n$ for information-content of gene j belonging to operon i.

Three intensities are associated with the first structure (C) and m_{ij} intensities associated with structure G_{ij}, where m_{ij} is the number of information codons in G_{ij}, each of them representing one of the 61 possible information codons. However, assuming that the time of transcription of any of the 61 codons is constant (a reasonable assumption, see Bobrow, et al [22]) and in view of the fact that these codons are undistinguishable from one another, from kinetic data[1] we can aggregate the intensities representing each of the 61 information codons into a single one whose information content is simply the time elapsed (one unit of time) in transcribing a codon. A similar argument convinces us that the two initial codons can be aggregated also into a single one.

The aggregation procedure for operons described in the last paragraph can be put into the formal framework of intensity homomorphic aggregation introduced previously, as follows.

Let τ be an operon type with structure set $\Phi_\tau = \{a,b,e,d, C, G_{i,1}, G_{i,2},\ldots,G_{i,n}\}$ where a,b,e,d are the structures associated with transcriptional control of the operon (see Section 3.3). C is the initiation-termination structure and $G_{i,1},\ldots,G_{i,n}$ are the information structures, one for each gene in the operon. The set of intensities Q_τ is given by

[1]This is the only data we are using because it is the only data that can be obtained while the process under study in evolving (see Rosen [13]).

$$Q_\tau = \{ON, OFF, s|by_1, s|by_2, READ, q_{1,1}, q_{1,2}, \ldots, q_{1,T_{i,1}}$$
$$q_{2,1}, \ldots, q_{2,2}, \ldots, q_{2,T_{i,2}}, \ldots, q_{n,1}, \ldots, q_{n,T_{i,n}}\}$$

where ON, OFF are intensities associated with the transcriptional control, $s|by$, $s|by_2$ represent the two termination codon and $q_{j,T_{k_i,k}}$ represent the information codons in the operon $q_{j,T_{k_i,k}}$ is the k-th codon in the jth-gene of operon i) where each $q_{j,T_{i,k_i}}$ is one of G_1 possible intensities.

We define the mapping $\phi_1 : Q_\tau \to Q_\tau$, where Q_τ, = {ON, OFF, $s|by$, READ, $q'_{1,1}, \ldots, q'_{n,T_{i,n}}$}, by

$$ON\ \phi_1 = ON$$

$$OFF\phi_1 = OFF$$

$$s|by_1\phi_1 = s|by_2\phi_1 = s|by$$

$$READ\phi_1 = READ$$

$$\{q_{i,T_{i,k_i}}\ \phi_1 = q'_{i,T_{i,k_i}}\}$$

with $q'_{1,1} = 1$, $q'_{1,2} = 2, \ldots, q'_{1,T_{i,1}} = T_{i,1}$,

$q'_{2,1} = 1$, $q'_{2,2} = 2, \ldots, q'_{1,T_{i,2}} = T_{i,2}$

$q'_{n,1} = 1$, $q'_{n,2} = 2, \ldots, q'_{1,T_{i,n}} = T$

we can see that the resulting τ' is an intensity homorphic aggregation of

of τ, and the congruence classes on Q_τ by the kernel of ϕ_1 are

$$\{\text{ON}\},\ \{\text{OFF}\},\ \{\text{READ}\},\ \{s|by_1,\ s|by_2\},\ \{q_{1,1},q_{2,1},\ldots,q_{n,1}\},\ldots,$$

$$\{q_{t,T_{i,1}},q_{t,T_{i,2}},\ldots,q_{t,T_{i,n}}\}$$

The resulting aggregated type for the operon τ/k sketched above is very close to the one described in Section 3.3. In fact, the only difference consists in that in Section 3.3 we split the operon type into two elements; 1) an element (OP) that carries the transcriptional control structure, the initiation-termination structure and a set of information structures, one for each gene in the operon with associated intensities representing the time of transcription of the nucleotides[1] of the respective gene and 2) an element (Cl) which counts elapsed time during transcription of the genes. Clearly, these two elements are operationally equivalent to the aggregated type discussed above.

We now continue with our general discussion of intensity homomorphic aggregation.

Notice that although the concept of intensity homomorphic aggregation was defined in the context of a single type τ, equations (4) and (5) imply that the epimorphism ϕ_1 must also induce a homomorphic aggregation on the intensity sets of the types that interact with τ in the informa-

---

[1]Since 3 nucleotides form a codon and since time in the model is a relative qunatity (i.e., with respect to the time horizon of the system under study, it is clear that it is equivalent to determine a quantity representing transcriptional time from the number of nucleotides of the genes or the number of codons. In our model, one step in time represents the reading of a codon (see discussion of the time of synthesis in Section 3.3).

tional space. Specifically, those types whose intensities are <u>inputs</u> to τ[1], have homomorphic images under ϕ_1. In addition, if the intensities of τ are the inputs to a type τ_1 in the informational space, $(q)C$ is the congruence class of Q_τ under the kernel of ϕ_1 containing $q \in Q$. Then, for every structure transition function of τ_1, $f^\gamma_{\tau_1} : \mathcal{D}_o f^\gamma_{\tau_1} \to Q_{\tau_1}$, we require the following <u>compatability</u> condition. Let τ be the ith-nearest neighbor of τ_1 in the informational space; let q^1, q^2 be any of two elements in $(q)C$ then

$$(q_1,\ldots,\underset{\substack{\uparrow\\ \text{ith position}}}{q^1},\ldots,q_{n_\gamma})f^\gamma_{\tau_1} = (q_1,\ldots,\underset{\substack{\uparrow\\ \text{ith position}}}{q^2},\ldots,q_{n_\gamma})f^\gamma_{\tau_1} \qquad (6)$$

Similarly, the compatability condition for the structure transition function H_{τ_1} of τ_1 becomes

$$(\gamma,\ (q_1,\ldots,\underset{\substack{\uparrow\\ \text{ith position}}}{q^1},\ldots,q_{n_\gamma}))H_{\tau_1} = (\gamma,\ (q_1,\ldots,\underset{\substack{\uparrow\\ \text{ith position}}}{q^2},\ldots,q_{n_\gamma})H_{\tau_1} \qquad (7)$$

$$\forall\ \gamma \in G_{\tau_1}$$

In the actual construction of the model of Section 3.3, we first aggregated the operon type as discussed above, and then proceeded to build the other types of the model according to the compatibility conditions (5) and (6). By this procedure we eventually obtained aggregated

---

[1] i.e., those types that pass information to τ.

types for the different classes of elements (see Section 3.1). Once we obtained aggregated types for the transformation elements, whose intensities are representations of concentrations of chemical components, which are the kinetic data available about the behavior of epigenetic processes (see Section 2.1) we checked if the level of aggregation in our transformation types is compatible[1] with this data. Since this in fact was the case, (see Chapter 5), we did not have to iterate in our aggregation procedure (see Section 3.1).

However, a glance into the literature of experimental biology (see for instance Sampson [ ]), convinced us that for many epigenetic processes under current study the available experimental kinetic data is far less complete than that assumed to exist and described in Section 3.1. In these cases, we could further aggregate the transformation element type so as to satisfy the following condition:

Let Q_τ be the intensity set of the transformation type and let K_c be the set of numbers on which the quantized and discretized version of the concentration evolution of the different elements of the process takes values. Let $h : Q_\tau \to K_c$ be the assigning map; if this map is not 1-1 and onto, we can reduce the size of Q_τ by replacing it with the equivalence classes induced on it by the congruence relation K_h defined

[1] By compatible we mean an affirmative answer to the following question: Given an epigenetic process can we reproduce the quantized and discretized version of kinetic data about the process with a simulation of it with our model?

by the kernel of h. Then we can proceed further to aggregate the types of the model according to this congruence relation. This procedure is still homomorphic aggregation because K_h, belongs to the sublattice of congruence relation of Q_τ which satisfy $K_h \geq K_\tau$[1] where K_τ is the congruence relation originally defined on the operon type and "propagated" to the other types of the model as described previously.

The procedure sketched above could be easily mechanized (e.g., as a computer program) this could give our model the added flexibility of being able to "adapt" itself to the level of aggregation of the kinetic experimental data available. We will not carry out this implementation in this thesis, but mention it here as a future improvement in the computer construction of our model which is described in the next section for several examples.

3.5 Examples

In this section we carry out an implementation of a model of the characteristics described in Section 3.3 for two epigenetic control processes. The first one, whose description will be given in subsection 3.5.1 is the well known lac operon (see Lewin [11], Watson [9], and Gutfreund [17]) of E. Coli. The second, described in subsection 3.5.2 is an abstract epigenetic system consisting of 3 operons, whose opera-

[1] See theorem A.1-2.

tional diagram was introduced in Section 2.1 (see Figure 2-1.6). The second example is a particular version of a class of epigenetic processes called self instructive catalytic hypercycles or Eigencycles and were proposed by Eigen [14], as an idealized version of many genetic control processes in procaryotes. In subsection 3.5.2 we shall examine some of its characteristics in the context of an implementation of our model.

Our main objective in this section is to describe in the context of the two examples mentioned above, a conversational program (called IDHA) for implementing epigenetic control systems with models of the type discussed in Section 3.3.

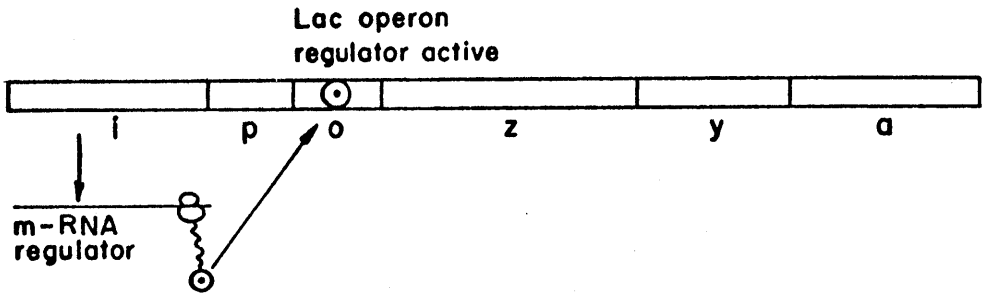
3.5.1 The lac operon

A schematic representation of the DNA of the lac operon is shown in Figure 1. As can be seen from the figure, this operon consists of 3 structural genes labelled z, y and a[1], a gene coding for the regulator protein labelled i.

The segments of DNA labelled p and o are the promoter and operator regions of the operon. They do not cade for proteins, but actually their purposes are to serve an an initial binding site for an m-RNA polymerase and binding site for the regulator protein respectively, as illustrated in Figures 2 and 3.

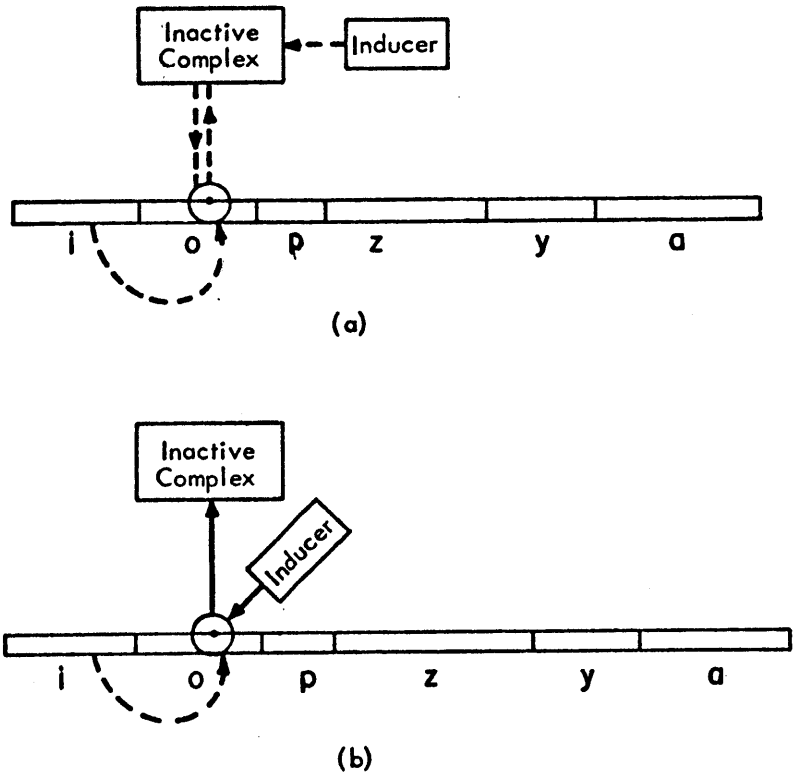
---

[1]These labels are the ones conventionally used in the literature (e.g., Watson [9]).



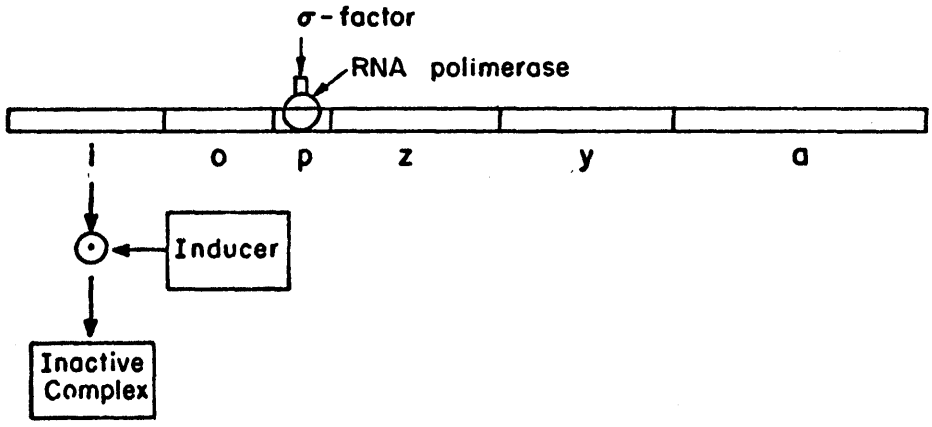
Lac Operon Inactive

Figure 1



Models for release of regulator protein from lactose DNA by inducer. (a) Allosteric modification of the regulator by the inducer; (b) Direct interaction of inducer and regulator protein in the operator region

Figure 2



Lac Operon Active

Figure 3

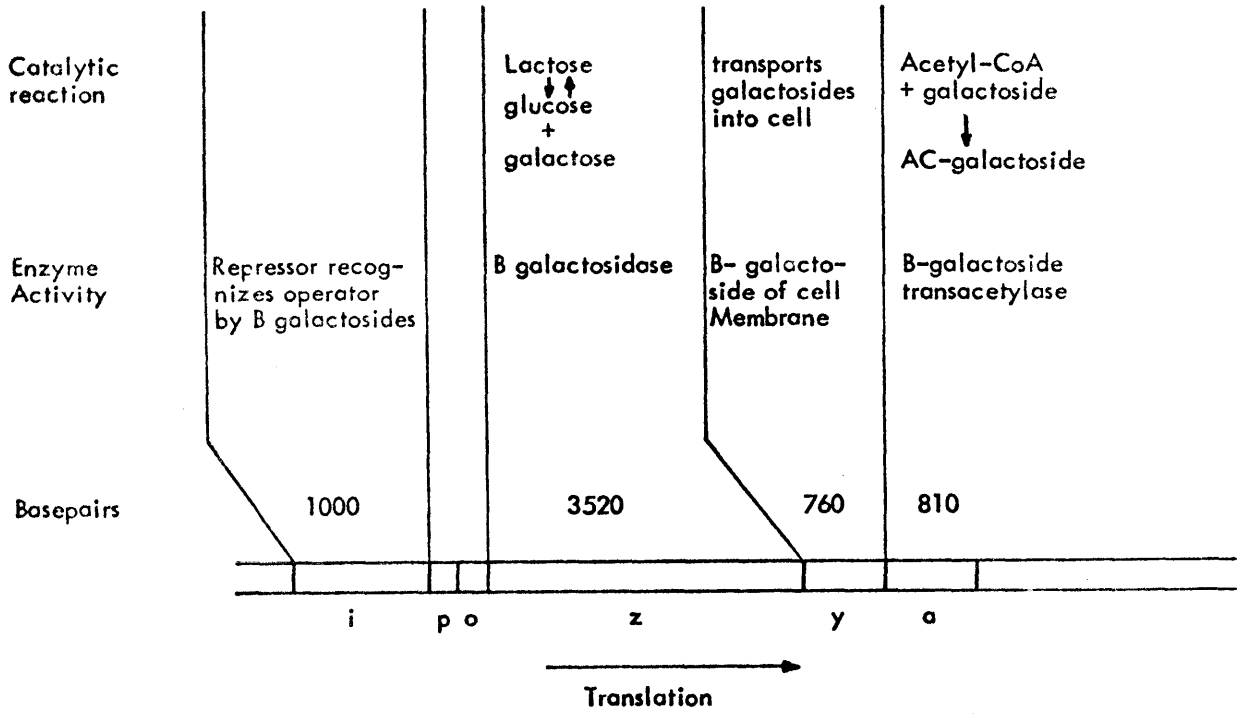


Figure 4

The lactose operon, the order of genes and control sites

Conventions: p = promoter
o = operator
i = control gene (see text)
z,y,a = structural genes of the operon

The number in the last line indicates estimated number of nucleotides (adapted from Lewin).

The control of the lac operon is of the negative control type, i.e., repressible (the regulator protein turns transcription off), and inducible (the control metabolite induces transcription). This type of control will be referred to as negative-inducible. Figure 2 illustrates two possible physical mechanisms for this control; in 2a the presence of the control metabolite (called inducer for this type of control) produces an allosteric modification[1] of the regulator protein such that it can no longer bind to the operator region. In Figure 2b, direct interaction between inducer and regulator protein at the binding site (the operator region as consequence of which the resulting complex is unable to bind to the operator region allowing the M-RNA polymerases to bind to the promoter region and proceed with the transcription of the operon. We note that from the operational point of view, both mechanisms are equivalent in the sense that the control actions of regulator protein and control metabolite produces effect which is schematically illustrated in Figure 3.

In Figure 4, a scheme of the lac operon and some of its operational characteristics are given, these are:

1 -- The length (estimated) of the genes of the operon

2 -- The control action of the corresponding proteins[2]

---

[1]This allosteric modification can be thought of as a competetive reaction involving the operator region, the regulator protein and the inducer with dominance of the inducer (see Reiner [15]).

[2]The exact control action, of the a-protein (β-galactoside-transacetylase) is not known.

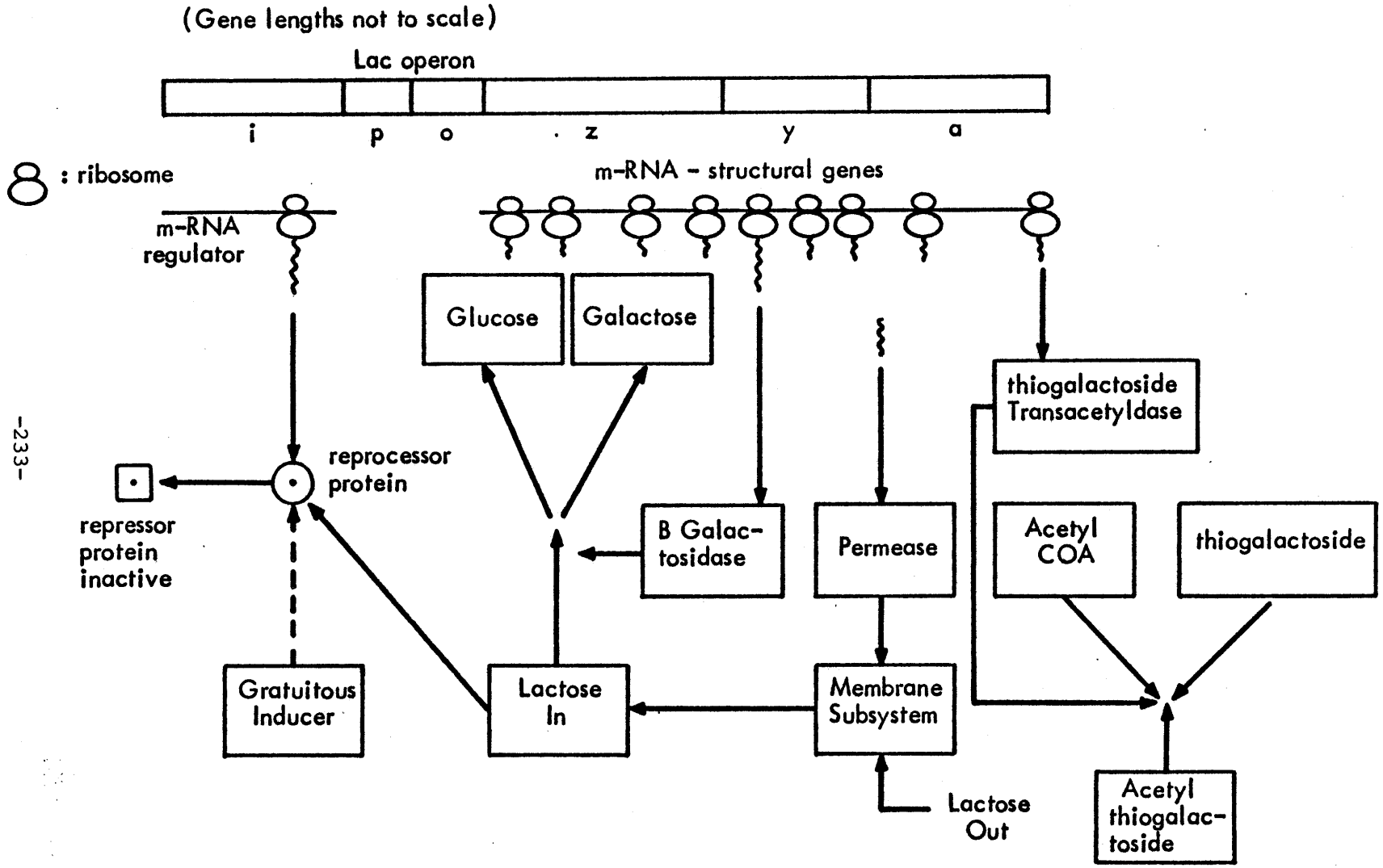


Fig. 5 Lac Operon in Active State

3 -- The reaction systems (transformation elements) associated (controlled by) each of the proteins.

These three items are the input data for the implementation of the distributed hierarchical model for the operon. (This implementation is carried out by IDHA in a conversational mode with the user. An example of its operation is shown in the printout shown in Table 1 for the lac operon implementation).

Figure 5 gives an incidence diagram of the dynamic interaction among the different elements participating in the lac-epigenetic process. The central task of IDHA is to transform the input data and the associate incidence diagram into specifications in terms of the types of our model.

The only element in the diagram of Figure 5 we have not discussed explicitly in the construction of the types of the model is the gratuitus inducer. Gratuitus inducers (repressors) are metabolites (see Watson [ ], Lewin [ ]) which have the ability to induce (repress) operons which are not metabolized by the reaction systems of these operons. The operational importance of these metabolites consists in that the number of elements present in the system remains constant during its evolution (assuming that they are not affected by any other operon in the system). We represent gratuitus inducers (repressors) in our model with elements of the transformation class of monomolecular irreversible type:

$$A_{i,k_i} \rightarrow A_{i,k_i}$$

where the symbol A_{i,k_i} represents the intensity assigned to the inducer.

Before discussing IDHA in the context of this example, we make the following important remark. Since the p and o regions of the DNA of the operon are not codings for proteins, but rather part of its control apparatus we do not model them as genes, but as part of the control logic of the transcription feedback mechanism (TFM), an element in our model) associated with the operon element. We add that the lengths[1] of o and p are usually very small as compared with the average lengths of the genes. Typically the ratio between the latter and the former ranges between 40 and 200.

Now, we proceed to discuss the characteristics of the program IDHA and illustrate them with the example considered in this subsection.

The first and second lines in Table 1, correspond to the loading and initialization of the program (IDHA) respectively.

Then, as can be inferred from the table, the program requests several items to the user in conversational mode (meaining that the execution halts after each request, and does not continue until it has been answered).

The decimal point that appears in the answers to most requests is used as a token and has no numerical meaning.

In the case of the lac operon, we have that the system has 1 operon,

[1]We understand as "length" in here, the number of nucleotides.

Table 1

IDHA Construction of the Types for
the Lac-Operon System

```
RESET("IDHA",110)

IDHA

ENTER NUMBER OF OPERONS
1.
ENTER NUMBER OF GENES OF EACH OPERON, ONE PER LINE
4.
ENTER ESTIMATED NUMBER OF NUCLEOTIDES FOR GENE NUMBER REQUESTED
 1 1
1000.
 1 2
3520.
 1 3
760.
 1 4
810.
ENTER MAXIMUM ERROR ALLOWED IN TIMES OF SYNTHESIS
.1
ENTER PROTEIN TYPES 1. FOR REGULATOR, 2. FOR UNIMOLECULAR IRREVERSIBLE
3. FOR UNIMOLECULAR REVERSIBLE, 4. FOR BIMOLECULAR IRREVERSIBLE,
5. FOR BIMOLECULAR REVERSIBLE, 6. FOR CYCLIC PROTEIN
 1 1
1.
 1 2
5.
 1 3
4.
 1 4
5.
FOR REGULATOR PROTEINS ENTER INDEX OF CONTROLLER OPERON AFTER EACH
PAIR OF SUBINDICES OF CORRESPONDING PROTEINS APPEAR
 1 1
1.

          STATE ASSIGNMENT FOR OPERON AND CLOCK ELEMENTS
 GENE STRUCTURE        NUCLEOTIDES        ERROR      T.O.SYNT.
 G(1,1)=1.                1000.           .0000      T(1,1)=10.
 G(1,2)=2.                3520.           .0057      T(1,2)=35.
 G(1,3)=3.                 760.           .0520      T(1,3)=8.
 G(1,4)=4.                 810.           .0100      T(1,4)=9.
```

Table 1 (contd.)

GENE STRUCTURE	INTENSITIES
C=7.	S/BY=36.
	READ=37.
A=8.	OFF=38.
	ON=39.
B=9.	
C=10.	
D=11.	
CLOCK STRUCTURE	INTENSITIES
0.	0.
COUNT=5.	T(1,1)=10.
	T(1,2)=35.
	T(1,3)=8.
	T(1,4)=9.
SYNTH=6.	TS(1,1=10.
	TS(1,2)=35.
	TS(1,3)=8.
	TS(1,4)=9.

MRNA,MRNA POL., RIBOSOME STRUCTURE/INTENSITY ASSIGNMENT

MRNA STRUCTURE	INTENSITY
N=16.	0.
END=17.	TS(1,1)=10.
	TS(1,2)=35.
	TS(1,3)=8.
	TS(1,4)=9.
MARK=18.	M=44.
	0.
DESTROY=19.	DE=45.
MRNA POL. STRUCTURE	INTENSITY
NOOP=12.	0.
ACT=13.	BR=40.
	S=41.
	R=42.
	0.
MOTION=14.	R=42.
	0.
UMOTION=15.	SET=43.
	0.

Table 1 (contd.)

RIBOSOME STRUCTURE	INTENSITY
INA(1)=20.	1.
INA(2)=21.	2.
INA(3)=22.	3.
ONE=23.	
TWO=24.	
THREE=25.	BR=40
	CONT=44.
	DECONT=45.
	S1=41.
	RR1=42.
TRANS=26.	

ENTER REACTANTS AND/OR PRODUCTS SHARED BY MORE THAN ONE REACTION SYSTEM
USE 1. FOR A, 2. for B, 3. FOR C
IN A+B=C OR A=B TYPES OF REACTIONS
ENTER IN ONE LINE THE TYPES OF ELEMENT FOLLOWED BY THE TUPLE IDENTIFYING THE OPERON AND PROTEIN CONTROLLING THE REACTION SYSTEM

3. 1. 2. 3. 1. 3.

REACTION SYSTEM STRUCTURE	INTENSITY
RA(1,2)=27.	A(1,2)=46.
RB(1,2)=28.	B(1,2)=47.
RC(1,2)=29.	C(1,2)=48.
LA(1,2)=30.	
LB(1,2)=31.	
LC(1,2)=32.	
RA(1,3)=33.	A(1,3)=49.
RB(1,3)=34.	B(1,3)=50.
RC(1,3)=29.	C(1,3)=48.
LA(1,3)=35.	
LB(1,3)=36.	
LC(1,3)=32.	
RA(1,4)=37.	A(1,4)=51.
RB(1,4)=38.	B(1,4)=52.
RC(1,4)=40.	C(1,4)=53.
LA(1,4)=41.	
LB(1,4)=42.	
LC(1,4)=43.	

Table 1 (contd.)

PROTEIN STRUCTURE	INTENSITY
NE=-4.	0.
SELECT=45.	T(1,1)=10.
	T(1,2)=35.
	T(1,3)=8.
	T(1,4)=9.
RR(1,1)=46.	REG(1,1)=54.
	P(1)=55.
LR(1,1)=47.	
RS'''(1,2)=48.	A(1,2)=46.
	B(1,2)=47.
	C(1,2)=48.
	AB(1,2)=56.
RS''(1,3)=49.	A(1,3)=49.
	B(1,3)=50.
	C(1,3)=48.
	AB(1,3)=57.
RS'''(1,4)=50.	A(1,4)=51.
	B(1,4)=52.
	C(1,4)=53.
	AB(1,4)=58.

ENTER A 1. IF OPERON I IS INDUCIBLE - NEGATIVE
ENTER A 2. IF OPERON I IS INDUCIBLE - POSITIVE
ENTER A 3. IF OPERON I IS REPRESSIBLE - NEGATIVE
ENTER A 4. IF OPERON I IS REPRESSIBLE - POSITIVE

ENTER CONTROL METABOLITE OF OPERON I (SEE INTENSITY NUMBER IN
TABLE OF REACTION-SYSTEMS INTENSITIES)
I= 1.
1.
48.

TFM STRUCTURE	INTENSITY
CP1(1)=51.	0.
	P(1)=55.
CM1(1)=52.	M(1)=48.
S1=53.	A=59.
	INA=60.

that the operon has 4 genes, whose lengths are entered as requested. We note that the program assigns the label 1.1 to the first length of the first gene in the first operon; the label $1,k_1$ to the length of the last gene of the operon; the label 2,1 to the length of the first gene of the second operon; and so forth. Thus, these must be entered in that order. Once they are entered, the gene 1.3,say, has been associated with a given length (760.) which will determine its time of synthesis.

Next, once all the lengths are entered a request for times of synthesis error has been satisfied, the program computes the times of synthesis of each protein according to the formula

$$T_{i,k_i} = [(L_{i,k_i}/\text{Iseed})] \tag{1}$$

where L_{i,k_i} is the length of the k_i gene of the i-th operon and Iseed is a scaling factor, initially assuming to be 100. The brackets [ ] indicate "integer-part function".

The error ε_{i,k_i} commited in assigning the time of synthesis of the k_i-th gene of the i-th operon according to formula (1) is expressed as follows:

$$\varepsilon_{i,k_i} = \frac{L_{i,k_i} - T_{i,k_i}\ \text{Iseed}}{L_{i,k_i}} \tag{2}$$

we demand that

$$\varepsilon_{i,k_i} < \varepsilon \quad \forall\ i,k_i \tag{3}$$

$$T_{i,k_i} \neq T_{j,k_j} \qquad j \neq i \qquad (4)^1$$

If condition (4) is not satisfied we make Iseed ← Iseed-5 and recompute the times of synthesis according to (1) and check conditions (3) and (4) again. If they are satisfied, we take the resulting set of times of synthesis as the times of synthesis for genes of the system under study, otherwise we decrease Iseed by 5. and repeat the computation until conditions (1) to (4) are satisfied or Iseed = 5. In the latter, we take Iseed equal to 1 and $T_{i,k_i} = L_{i,k_i}$ $\forall$ i.

We note that Iseed >1, implies that we save in the number intensities required for representing the clock of the model, and we compress the simulation time by a factor proportional to Iseed.

Once the gene lengths have been entered the program assigns the tuple (i,k_i) to protein i,k_i and requests protein-type for each (i,k_i) as indicated in Table 1. Finally, for regulator proteins the program requests for each one, the label (number) of the operon controlled by it.

Once the above information is entered, the program proceeds to assign integers to the different structures and intensities of the model corresponding to the system under study, using the following criterion:

a) The number 0 is reserved for the dormant structure and the dormant intensities.

---

[1] For instance, in the computation of T(1,3), T(1,4), in the first iteration we get T(1,3) = T(1,4) = 8 since these times are the identifiers of the genes, we need them to be different, thus, in the second iteration we obtain T(1,3)=8, T(1,4)=9.

b) The numbers $1,2,\ldots,T_{i,k_i}$, $T_{i,k_i+1},\ldots$, are used as the intensities of the clock in structure COUNT (see Section 3.3).

c) The numbers assigned to structures of the operon type are computed as follows

$$G_{i,k_i} = i + k_i - 1 \tag{5}$$

Let

$$S' = \max_{i,k_i} G_{i,k_i} \tag{6}$$

The structure C' is assigned[1] the number

$$C' = S' + 3 \tag{7}$$

and the structures A,B,C,D are assigned the numbers

$$A = C'+1,\ B = C'+2,\ C = C'+3,\ D = C'+4 \tag{8}$$

For instance, in the example of the lac operon displayed in Table 1

$S' = 4$

$C' = 7$

$A = 8$

$B = 9$

$C = 10$

$D = 11$

---

[1]The labels used in this section for identifying structures are the ones established in Section 3.3.

d) The number assigned to the intensity of the structure G_{i,k_i} is T_{i,k_i}, the corresponding time of synthesis of Pr_{i,k_i}.

e) The clock structure COUNT is assigned the number

$$COUNT = S' + 1 \quad \text{(e.g., COUNT = 5 in Table 1)} \tag{9}$$

and the structure SYNTH is assigned the number

$$SYNTH = S' + 2 \tag{10}$$

f) The intensities of structure SYNTH $TS(i,k_i)$ are assigned numbers according to the formula

$$TS(i,k_i) = T_{i,k_i} \tag{11}$$

g) Let U' be a number defined by

$$U' = \max_{i,k_i} T_{i,k_i} \tag{12}$$

(e.g., U' = 36 in the example of Table 1)

The number assigned to intensities S|by, READ associated with structure C' of the operon element are:

$$S|by = U' + 1 \tag{13}$$

$$READ = U' + 2$$

while the numbers assigned to intensities ON and OFF associated with its structures A,B,C,D are

$$ON = U' + 3$$
$$OFF = U' + 4 \qquad (14)$$

h) Set

$$U' \leftarrow U' + 9^{1}, \quad S' \leftarrow S + 6 \qquad (15)$$

Assign to structures N and END of the M-RNA element the numbers S' and S'+1 respectively, and S'+2 and S'+3 to its structures MARK and DESTROY, respectively.

i) Assign numbers U' and U'+1 to intensity M of structure MARK and to intensity DE of structure DESTROY, respectively (44, 45 in Table 1).

j) Set U' ← U' - 4, S' ← S' - 3. Assign numbers S', S' +1, S' +2, S' +3 to structures NOOP, ACT, MOTION, respectively and numbers U', U' +1 and U' +2 to intensities BR, S and R respectively of the M-RNA element (12, 13, 14, 15 for the corresponding structures in Table 1, and 40, 41, 42 for the intensities).

k) Assign numbers S' +4 and S' +5 to structures MOTION and UMOTION of the M-RNA polymerase structure and the numbers U' +3 and U' +4 to their intensities R and SET respectively (14, 15 for the corresponding structures in Table 1, and 42, 43 for the intensities).

l) Set U' ← U' +10, S' ← S' +7. Assign the numbers S', S' +1 and S' +2 to structures INA(1), INA(2) and INA(3) of the ribosome element,

and numbers S' +3, S' +4, S' +5 to its structures ONE, TWO and THREE. (23, 24 and 25 respectively in Table 1).

m) Assign numbers U' and U' +1 to intensities CONT and DECONT of the structures ONE, TWO and THREE. (44 and 45 respectively in Table 1).

n) The assignment for structures and intensities of reaction systems depends on the type of each reaction system (which is determined in the program once the request for protein types is answered, as shown in the sample-output of IDHA for the lac operon) and also, on the information (supplied by the user) as to what molecules belong to two or more reaction systems.

Let N_{i,k_i} be the reaction type of system $R(i,k_i)$ or rather the protein-type controlling reaction $R(i,k_i)$ so (see Table 1)

$$N_{i,k_i} = \begin{cases} 2 \text{ or } 3 \text{ if } R(i,k_i) \text{ is a monomolecular} \\ \text{reaction system} \\ 4 \text{ or } 5 \text{ if } R(i,k_i) \text{ is a bimolecular} \\ \text{reaction system} \end{cases} \tag{16}$$

Let S', U' be numbers assigned to the last structure and intensity (respectively) considered up to this point in the execution of IDHA.

If $N_{i,k_i} = 2$ or 3 and A_{i,k_i} (the reactant, see Sections 3.1 and 3.3) and B_{i,k_i} (the product) are not shared by any other reaction $R(j,k_j)$ $j<i$ or $j=i$ and $k_j < k_i$, then set

$$\begin{aligned} Rr(i,k_i) &= S' +1 \qquad & A(i,k_i) &= U' +1 \\ R\ell(i,k_i) &= S' +2 \qquad & B(i,k_i) &= U' +2 \end{aligned} \tag{17}$$

If either A_{i,k_i} (or B_{i,k_i}) is shared with some reaction system $j<i$ or $j=i$, $k_j < k_i$ and $A_{j,k_j} = L$ (or $B(j,k_j) = M$), set

$$\begin{aligned} Rr(i,k_i) &= S' + 1 & A_{i,k_i} &= L \ (B(i,k_i) = M \\ R\ell(i,k_i) &= S' + 2 & B_{i,k_i} &= U' + 1 \ (A_{i,k_i}) = U' + 1 \end{aligned} \tag{18}$$

If both A_{i,k_i} and B_{i,k_i} are shared with some reaction system $R(j,k_j)$ $j \leq i$ (or $j=i$, $k_j < k_i$) and $A(j,k_j) = L$, $B(j,k_j) = M$, $R(j,k_j)$ is of the same type as $R(i,k_i)$ and $Rr(j,k_j) + N$, $R\ell(j,k_j) = P$ then set

$$\begin{aligned} Rr(i,k_i) &= N & A(i,k_i) &= L \\ R\ell(i,k_i) &= P & B(i,k_i) &= M \end{aligned} \tag{19}$$

If $N_{i,k_i} = 3$ or 4, a similar set of conventions are used for assigning numbers to the structures $RA(i,k_i)$, $RB(i,k_i)$, $RC(i,k_i)$, $LA(i,k_i)$, $LB(i,k_i)$, $LC(i,k_i)$ and to the intensities $A(i,k_i)$, $B(i,k_i)$, $C(i,k_i)$. Some of these conventions are used in constructing the number assignments for the reactions systems of the lac operon shown in Table 1. For instance, the reaction systems R(1,2) and R(1,3) share their C-elements i.e., $C(1,2) = C(1,3) = 48.$, where C(1,2) or C(1,3) represent the lactose molecule (see Figure 5 and Table 1).

p) Set $S' = \max_{i,k_i} \Sigma_{i,k_i}$ where Σ_{i,k_i} is Rr_{i,k_i} or $R\ell_{i,k_i}$, or RA_{i,k_i}, or RC_{i,k_i}. Similarly set $U' = \max_{i,k_i} \Theta_{i,k_i}$ where Θ_{i,k_i} is A_{i,k_i}, or B_{i,k_i},

or C_{i,k_i}.

Clearly, S', U' are the last numbers assigned to a structure and an intensity so far.

q) Assign S' +1 to structure NE of the protein type and S' +2 to its structure SELECT. The intensities associated with NE and SELECT are 0, and $1,2,\ldots,T_{i,k_i},\ldots,T_{m,k_m}$, respectively whose assignments were carried out previously.

r) Set S' ← S' +3. If $N^{1}_{1,1} = 1$ set

$$\begin{aligned} &RR(1,1) = S' && LR(1,1) = S' +1 \\ &REG(1,1) = U' && P(1) = U' +1 \end{aligned} \tag{20}$$

where $RR(i,k_i)$, $LR(i,k_i)$ are structures of a regulator protein element and $REG(i,k_i)$, $P(i)$ are their intensities (see Section 3.3).

If $N_{1,1} = 2$, set $RS(1,1) = S'$ where $RS(i,k_i)$ is the structure of a monomolecular irreversible type, we note that in this case, numbers have already been assigned to the corresponding intensities.

$$\begin{aligned} &\text{If } N_{1,1} = 3 \text{ set } RS'(1,1) = S' \\ &\text{If } N_{1,1} = 4 \text{ set } RS''(1,1) = S' \\ &\text{If } N_{1,1} = 5 \text{ set } RS'''(1,1) = S' \\ &\qquad\qquad \text{and } AB(1,1) = U' \end{aligned} \tag{21}$$

---

[1] N_{i,k_i} was introduced in step n) (see (16))

$$\text{If } N_{1,1} = 6 \quad \text{set} \quad \begin{aligned} RA(1,1) &= S' \\ RB(1,1) &= S' + 2 \\ RD(1,1) &= S' + 3 \end{aligned} \tag{22}$$

where $RA(i,k_i)$, $RB(i,k_i)$, $RC(i,k_i)$, $RD(i,k_i)$ are the structures of the cycle protein type (see Section 3.3, Figure 3.3-10f and companion discussion)

$$\text{set} \quad \begin{aligned} BD(1,1) &= U' \\ CA(1,1) &= U' + 1 \\ AD(1,1) &= U' + 2 \\ BC(1,1) &= U' + 3 \end{aligned} \tag{23}$$

We note again, that other intensities associated with the structures in (22) or (23) have been assigned numbers previously.

Once the assignment of numbers to the structures and intensities of PR(1,1) is completed, the program proceeds to the assignment of numbers to the structures and intensities of protein PR(1,2) (if there is a PR(1,2) otherwise to protein PR(2,1)), PR(1,3),...,PR(m,k_m). The program assigns new numbers only to those structures and intensities that are particular to PR(1,2), PR(1,3),...,PR(m,k_m) (for instance, if $N_{1,2}$=5, then the only intensity to which no number has been assigned is AB(1,2)). Note that the structures Ne and Select are common to all the protein elements in the system.

s) Set $S' = \max_{i,k_i} \Sigma_{i,k_i}$ where Σ_{i,k_i} is $RR(i,k_i)$, or $LR(i,k_i)$ or $RS(i,k_i)$, or $RS'(i,k_i)$, or $RS''(i,k_i)$, or $RS'''(i,k_i)$, or $RA(i,k_i)$, or

$RB(i,k_i)$ or $RC(i,k_i)$ or $RD(i,k_i)$.

Set $U' = \max_{i,k_i} \Sigma_{i,k_i}$ where Σ_{i,k_i} equals $AB(i,k_i)$ or $BD(i,k_i)$, or $CA(i,k_i)$ or $AD(i,k_i)$ or $BC(i,k_i)$.

t) Let m be the number of operons of the system under study and $\alpha(I)$ $I=1,\ldots,m$ a set of variables, each one associated with any operon whose value is entered by the user upon request (see Table 1).

For $I=1,\ldots,m$

$\alpha(I) =$ 1. If operon I is inducible-negative[1*]

2. If operon I is inducible-positive

3. If operon I is repressible-negative

4. If operon I is repressible-positive

(24)

Set $A = U' + 1$

$INA = U' + 2$

where A and INA are intensities of the structures S^1, S^2, S^3, S^4 of the transcription feedback mechanism element (TFM) (see Figure 3-3.15) and companion discussion).

For $I=1$ if $\alpha(1) = 1$ set $CP^1(1) = S' + 1$

$CM^1(1) = S' + 2$

$S^1 = S' + 3$

(25)

[1]By inducible-negative we mean that the corresponding transcription feedback mechanism is induced by the presence of the control metabolite and turned off by the regulator protein (see Section 3.3, Figure 3.3-5 and companion discussion).

*In Table 1 $\alpha(1) = 1$.

$$\text{if } \alpha(1) = 2 \quad \text{set} \quad CP^2(1) = S' + 1$$
$$CM^2(1) = S' + 2 \qquad (26)$$
$$S^2 = S' + 3$$

$$\text{if } \alpha(1) = 3 \quad \text{set} \quad CP^3(1) = S' + 1$$
$$CM^3(1) = S' + 2 \qquad (27)$$
$$S^3 = S' + 3$$

$$\text{if } \alpha(1) = 4 \quad \text{set} \quad CP^4(1) = S' + 1$$
$$CM^4(1) = S' + 2 \qquad (28)$$
$$S^4 = S' + 3$$

Once the structures of TFM(1) have been assigned their labels (numbers) IDHA proceeds to the assignment of labels to the structures of TFM(2),...,TFM(m) in a similar manner with the provisto that if S^1 or S^2 or S^3 or S^4 have already been assigned labels, these are used for every instance in which they are required.

u) Set $U' \leftarrow U' + 3$. At this point the user is requested to enter, for each operon, the intensity of the corresponding control metabolite. He can do so by consulting the Assignment of Reaction Systems which has been printed out (by IDHA) earlier (see Table 1).

v) Set $S' = \max_i \Sigma_i$ where Σ_i is CP(I), or CM(I), or S^1, S^2, S^3, or S^4

set STOP = S' +1

REG = S' +2

where STOP and REG are the structures of the corresponding virtual elements introduced in Section 3.3.

Step (v) finalizes the number assignment to structures and intensities of the elements of our model representing a given system provided by the user. We remark that the resulting model, constructed by IDHA, will depend on the characteristics of the system under study. For instance, in the case of the lac-operon, no monomolecular reaction systems are present; therefore, the corresponding structures of these protein and reaction-system types are absent.

We conclude our discussion of IDHA with a description of the way the resulting model is internally represented in the computer.

The model is represented in the computer by 3 arrays: {TYPES(I,J,K)} {TYPEI(I,J,K)} and {GN(J)}.

{TYPES(I,J,K)} is an array whose entries are determined by the structure transition functions of the different types of the model (see Section 3.3). {TYPEI(I,J,K)} is an array whose entries are determined by the intensity transition functions of the different types of the model and {GN(J)} associates with each structure of the model a subset of numbers from {0,1,...,G} representing the nearest-neighbors associated with that structure when the corresponding element has been allocated to the informational space. The organization of these arrays is illustrated in Tables 2, 3 and 4 respectively.

Initially, the 3 arrays are filled with 0 in each entry. One IDHA has assigned numbers to the different structures and intensities of the

		Present Neighbors Present Intensity	PRESENT STRUCTURE	NEXT STRUCTURE	
		J=0,1,...,6	7	8	
I=1	OPERON TYPE STRUCTURE TRANSITION FUNCTION	TYPES(1,J,k)			k= 1 2 . . . 26
		J=0,1,...,6	7	8	
I=2	CLOCK TYPE STRUCTURE TRANSITION FUNCTION	TYPES(2,J,k)			k= 1 2 . . . 100
		J=0,1,...,6	7	8	
I=3	MRNA POL STRUCTURE TRANSITION FUNCTION	TYPES(3,J,k)			k= 1 2 3 . . 14
.	.	.			
.	.	.			
.	.	.			

Table 2

Illustration of the Organization of the Array TYPES(I,J,K)

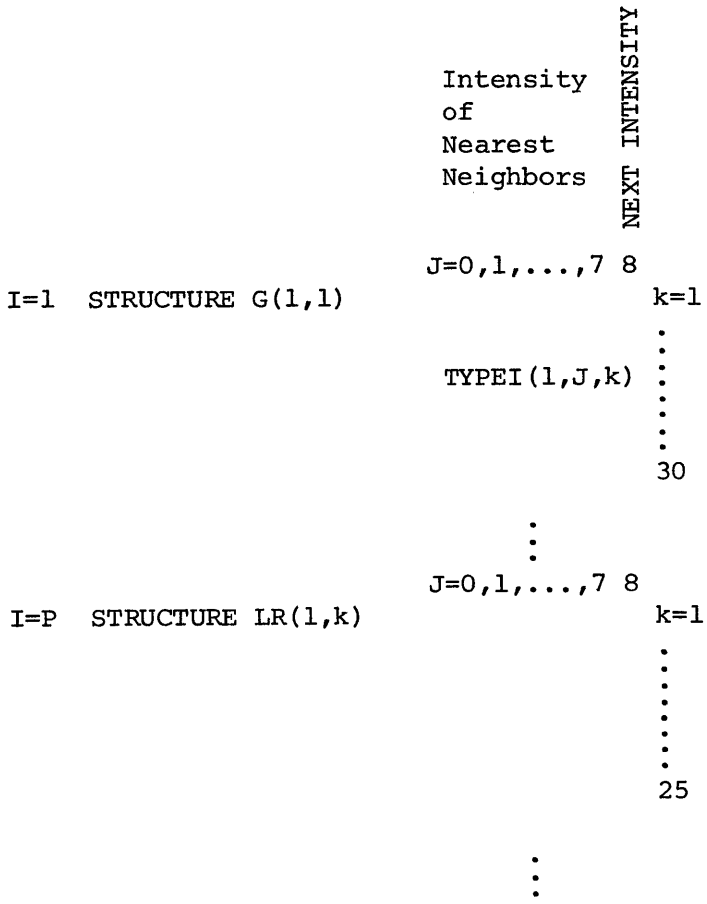


Table 3

Illustration of the Organization of the Array

TYPEI(I,J,K)

STRUCTURE (J)	0 1 2 3 4 5 6
	0... 1,..., 0

GN(J,ℓ)

$$GN(J,\ell) = \begin{cases} 0 \text{ if } \ell\text{th nearest neighbor does not interact with element when in J structure} \\ 1 \text{ otherwise} \end{cases} \qquad \ell=0,\ldots,6$$

Table 4

Illustration of the Array GN(J,ℓ)

model for the system under study, corresponding entries of these arrays are changed to the assigned numbers via a set of identities of the form

```
TYPES(1,7,1) = G(1,1)
TYPES(1,7,2) = G(1,2)
        :
        :
        :                                                    (29)
TYPES(1,7,M) = G(L,P)   M=L+P
TYPES(1,1,0) = T(1,1)
TYPES(1,1,1) = 1
        :
        :
   Etc...
```

where $G(I,k_i)$, $T(I,k_i)$ etc... are the variables representing structures or intensities defined in Section 3.3 and to which IDHA has assigned numbers on a 1-1 fashion. Notice that the first entry in the triplet I,J,K in TYPES(I,J,K) identifies the type while the third identifies the nearest neighboring element $0 \_ J \_ 6$ to which the corresponding intensity belongs. The entries in the columns J=7, J=8 correspind to the columns "present structure" and"next structure" in the appropriate table of Section 3.3.

In the array TYPEI shown schematically in Table 3 the entries are constructed from the intensity transition functions developed in Section 3.3. In TYPE(I,J,K) I idnetifies the structure and J,K identify the "coordinates" of the entry in the corresponding table, as illustrated in Table 2.

Finally, the array GN(J) determines for each structure J the corresponding nearest neighbors: if GN(J,) = 1, R = 0,1,...,6 this nearest neighbor is active for this structure, as indicated in Table 4.

We conclude this subsection with an observation of the model for the lac-operon which served us as a running example for introducing IDHA.

We note that the functional objective of the permease protein is to facilitate (catalyze) the flow of the lactose molecule from the outside to the inside of the cell. In our model we have represented this functional objective by modeling the lactose out as an operationally different molecule than the lactose IN, the membrane as an element of the transformation class and the lactose permeation itself as a bimolecular irreversible reaction of the form

$$\text{Lactose-OUT} + \text{Membrane} \xrightarrow[\text{Permease}]{} \text{Lactose-IN} \qquad (30)$$

This shows an instance in which our model can be used to study epigenetic actions which are not reacting in the strict sense of the word.

3.5.2 A 3-operon Eigen cycle

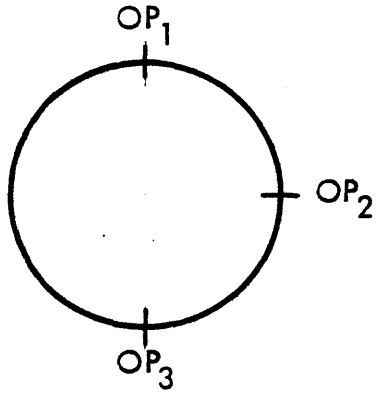
In this subsection we examine, in the context of the 3-operon example introduced in Section 2.1, some additional characteristics of the program IDHA which were not explicitly discussed in the last section.

The first of these concerns the physical locus of the operons in the cell. In general, two or more operons which interact via their

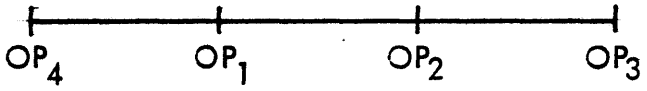
proteins and/or reaction systems as discussed in Section 2.1, are not spatially allocated in adjacent loci in the chromosome; therefore, in order to preserve time relations in the dynamics of the system composed of these operons we must, in the implementation of the model, provide it with the means to simulate the spatial distance between the operons of the system. This is accomplished, as shown in Figure 1, by allocating 2_n-1 operons to the informational space; for n operons in the system; where the additional n-1 operons are used to simulate the distance among the different operons. Each of the added operons is assumed to have a single gene which codes for its regulator protein and whose time of synthesis is proportional to the length between consecutive operons.

If OP^i is the "timing" operon added after OP_i, then OP^i has the same control metabolite M_i and regulator protein P_i, as OP_i.

The importance of Eigen cycles in epigenetic processes is the fact that in addition to the "local" transcriptional and catalytic controls associated with a given operon OP_i, there is a global control of the system due to the structure of the arrangement. The real cell locus of the chromosome is circular and it is clear that this organization favors the interdependence operons in a cycle. We <u>simulate</u> this circularity around a <u>given</u> operon OP_i (see Figure 2) in our model by creating additional copies of operons so that OP_i "sees" them in circular fashion. This is illustrated in Figure 6 for operon OP_i.



(a) 3 Operon allocation in the chromosome

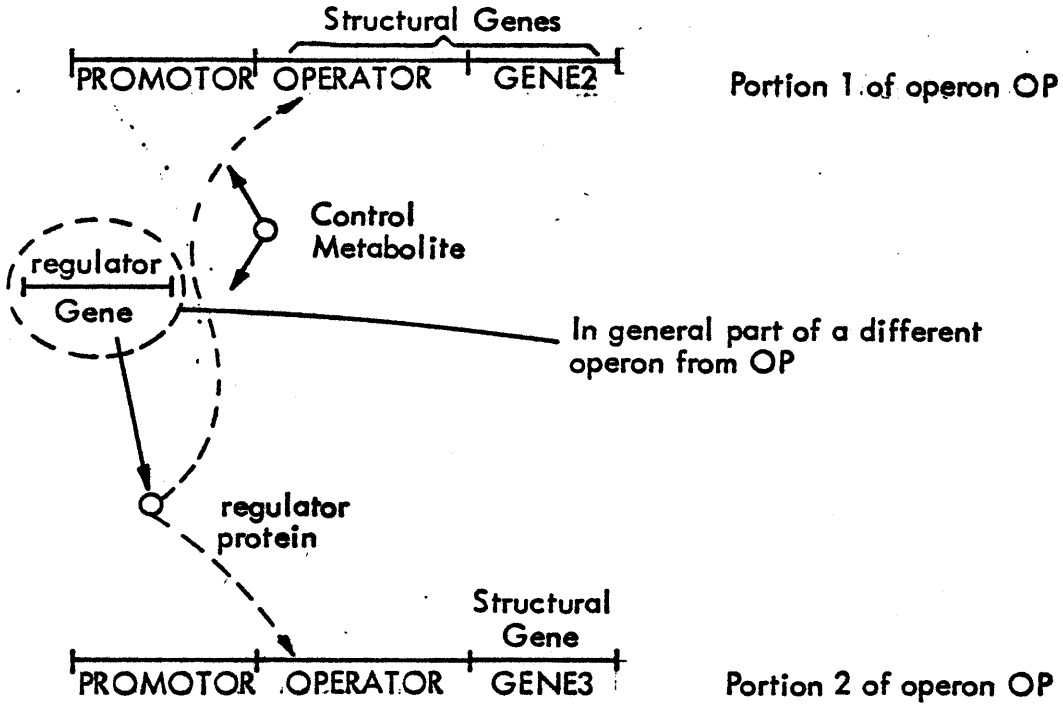


$OP_4 = OP_2$

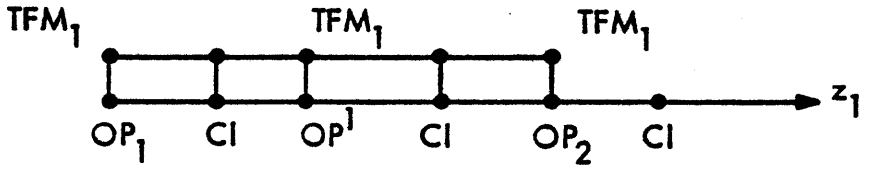
(b) Circular representation of (a) around OP_1

Figure 6

OPERON OP: 3 Structural Genes, GENE1, GENE2, GENE3



(a) Gene organization of a typical operon



(b) Representation of operon of part (a) in our model OP_1 represents Portion 1, OP_2 represents Portion 2 and OP^1 represents "timing" operon

Figure 7

In Figure 6a OP_3 and OP_2 are both in sequence of OP_1 because of the circular arrangement of the operons. This is simulated in our model by creating a new operon OP_4 with identical structure to that of OP_3 and allocate it as shown in Figure 6b if in addition we want to represent the circularity of OP_3 we add a new operon $OP_5 = OP_1$ at the right and on Figure 6b and a new operon $OP_6 = OP_2$ at the left end. If the time horizon of the system T_H satisfies the inequality

$$T_H \leq \sum_{i=1}^{m} \max_{i,k_i} \{T_{i,k_i}\} \tag{31}$$

where m is the number of operons in the model. We continue the procedure indicated above i.e., adding "neighboring" operons at the two ends of our operon string until the direction of inequality (31) is reversed; that is, the number of operons present for simulating circularity equals pt where p is the original number of the operon in the system and t is an integer chosen so that

$$T_H \leq \sum_{i=1}^{pt} \max_{i,k_i} \{T_{i,k_i}\} \tag{32}$$

In IDHA, there is a subroutine cycle (I) which when requested by the user generates the additional operons necessary for simulating circularity around operon I of the given system.

We close this subsection with the description of two additional features of IDHA in connection with the representation of multioperon systems.

We recall that an operon OP is a set of genes under common transcriptional control. This does not mean that all the genes of a given operon are spatially allocated in consecutive positions of the chromosome. As illustrated in Figure 3, we simulate this situation by creating one operon element for each consecutive portion of genes, with all these operons interacting with identical TFM elements as illustrated in Figure 3b.

Finally, in the cell there are operons which possess more than one transcriptional control (e.g., Ara in E coli)[1]. We simulate this in IDHA by "creating" one operon for each transcriptional control.

These last two operations -- operon "splitting" and TFM "splitting" -- are carried out in IDHA for any operon OP_i by a call to subroutines SPLIT(i) and TSPLIT(I).

---

[1]See Watson [ ].

4. FORMAL MODEL FOR EPIGENETIC CONTROL MECHANISMS: THE DISTRIBUTED HIERARCHICAL AUTOMATON

4.1 Formulation of the Model

In this chapter we develop a mathematical model for the dynamics of a computing device called the Distributed Hierarchical Automaton (DHA for short).

Although our purpose for the development of such a model is to obtain an efficient tool for analyzing the dynamics of the operational model established in Chapter 3, the DHA is an interesting computational model in its own right; therefore, in this chapter we study the DHA as an abstract dynamical system with only occasional references about its use in our study of epigenetic control processes.

In Chapter 5 we will develop a representation of the operational model of Chapter 3 as a DHA, and on the basis of this representation we formulate computational algorithms for the structural identification problem outlined in Chapter 2 and the three prototypes of epigenetic control processes stated in Section 2.3.

A distributed hierarchical automation A is a composite of two objects; I and τ called the Information space and the Type of A, respectively.

The information space I in this study is represented by the abelian group $(Z^k, +)$, where Z is the set of integers. $k \geq 1$ is an integer called

the dimension of I, and + is the ordinary addition on Z^k.

Occasionally, we will require a metric structure on I to establish properties of the DHA $\mathcal{A}$ that depend on the distance between points in I. This structure is provided in terms of the city-block metric function $\rho : Z^k \times Z^k \to Z$. We define ρ by the following formula. Let $z^1 = \{z_1^1,\ldots,z_k^1\}$, $z^2 = \{z_1^2,\ldots,z_k^2\}$ be two elements in Z^k. Then,

$$(z^1, z^2)\rho = \sum_{j=1}^{k} |z_j^1 - z_j^2| \tag{1}$$

where $|u|$ denotes the absolute value of u.

A type τ is a finite-state state output automaton. We assume that at each point in I, one and only one of these automata has been allocated.

τ consists of the following pieces of data:

(i) A set $Q = \{0,\ldots,R\}$ of natural numbers called the set of <u>intensities</u> of the type.

(ii) A set Φ of <u>indexed partial functions</u>;

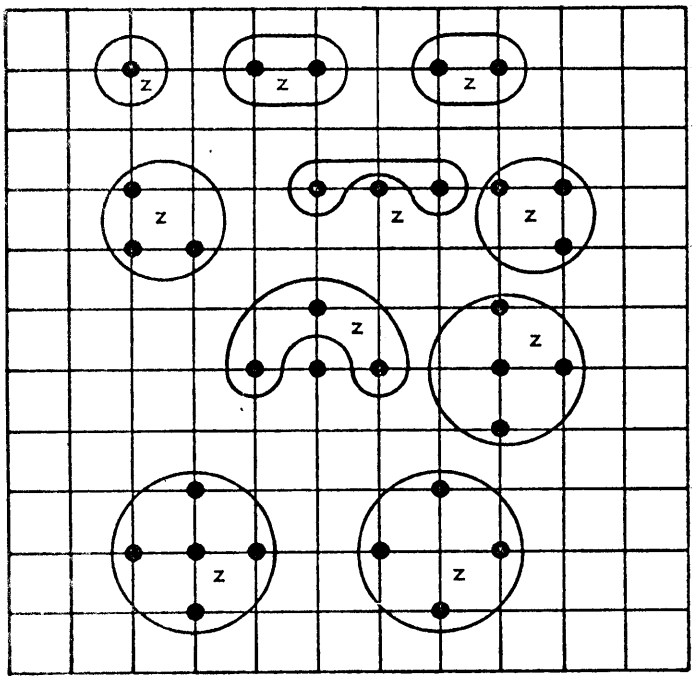
$$\Phi = \{f^\gamma | f^\gamma : \mathcal{D}_o(f^\gamma) \to Q \quad \forall\, \gamma \in G\} \tag{2}$$

called the set of structures of the type.

In (2), the set G is a finite set of natural numbers $\{0,\ldots,P\}$ called the set of labels of Φ. Further the map $\gamma \to f^\gamma$ is 1-1.

In (2), the sets $\mathcal{D}_o(f^\gamma)$ for each $\gamma \in G$, are defined as follows:

$$\mathcal{D}_o(f^\gamma) = \{ w \in \hat{Q}^{2k+1} | \ |w| = n_\gamma\} \tag{3}$$



Several Possible Neighborhoods of z in a Two-dimensional Informational Space

Figure 1

where $\hat{Q}^{2k+1}$ is the set of all sequences of elements of Q with at most 2k+1 elements, n_γ is the number of arguments of f^γ and $|w|$ is the length of w. Also, 2k+1 is the maximum number of nearest neighboring points of any point z in Z^k, including z itself. (A point $z^1 \in Z^k$ is a nearest neighboring point of $z \in Z^k$ if $(z, z^1)\rho \leq 1$).

We note that since f^γ is a partial function $\forall\ \gamma \in G$, the domain of f^γ is, in general, a possibly proper subset of $\mathcal{D}_o(f^\gamma)$.

(iii) A set of functions,

$$N = \{g_i,\ i=1,\ldots,\ell \mid g_i : Z^k \to 2^{Z^k}\} \tag{4}$$

where $\ell \leq 2^{2k}$, called the set of neighboring functions where each g_i maps a point of the information space into a finite subset of its nearest neighboring points (2^{Z^k} denotes the set of all subsets of points of Z^k). We require g_i's to be translation-invariant, in that they give the same shaped neighborhoods for each point.

An explicit representation for such a neighborhood function $g_i \in N$ is given by the formula

$$(z)g_i = \{z + \alpha_{i1},\ z + \alpha_{i2},\ldots,z + \alpha_i m_i\} \tag{5}$$

where $\alpha_{ij} = [\alpha_{ij}^1,\ldots,\alpha_{ij}^k]$ $\alpha_{ij}^s \in Z$ $s=1,\ldots,k$ are m_i distinct elements in Z^k, each one with all their entries equal to zero except, perhaps, one and only one of them. The nonzero entry (if any) can be either 1 or -1 and nothing else. Figure 1 illustrates the concept of neighborhood functions for a 2-dimensional information space. The number m_i of

elements $\{\alpha_{ij}, j=1,\ldots,m_i\}$ associated with a given function g_i is called the characteristic of the neighborhood function g_i.

We note that for any i, i=1,..., and every $z \in Z^k$, $(z)g_i$ denotes a subset of the 2k+1 possible nearest neighboring points of z. A simple combinatorial argument shows that there are 2^{2k} possible subsets of this kind.

(iv) A function $\Psi : G \to N$ called the Input selector function. Ψ is 1-1 and onto. The function Ψ satisfies the following consistency condition:

$$\forall \gamma \in G, \text{ for any } z \in Z^k, |(z)(\gamma)\Psi| = n_\gamma \tag{6}$$

(v) A function $H : G \times \hat{Q}^{2k+1} \to G$ called the structure transition function.

For each $\gamma \in G$, the function $H(\gamma, \circ) : \hat{Q}^{2k+1} \to G$ satisfies

$$\text{Domain } (H(\gamma, \circ)) \supseteq \mathcal{D}_o f^\gamma \tag{7}$$

For any $w \in \hat{Q}^{2k+1} - \mathcal{D}_o f^\gamma$, and any $\gamma \in G$,

$$H(\gamma, w) = 0 \tag{8}$$

where 0 is the label of a special structure, called the dormant structure which will be discussed later.

(vi) An arbitrarily large, but finite, subset of the non-negative integers $T = \{0,1,2,\ldots\}$ with the usual linear ordering ($<$) of the natural numbers, called the time base of the type.

The local dynamics of $\mathcal{A}$, i.e., the dynamics of τ can now be defined when we enbed this object in a $I = Z^k$; specifically let τ be the type of $\mathcal{A}$ allocated at $z \in Z^k$ (we indicate this allocation by $(z)\tau$); the input $w_t \in \hat{Q}^{2k+1}$ at time t to $(z)\tau$ is provided by a subset of the intensities of the types allocated in the nearest neighboring points of $(z)\tau$, at time t. This subset is determined by the input selector function of τ as follows.

Let $\gamma_t \in G$ be the label of $(z)\tau$ at time t, suppose

$$(\gamma_t)\Psi = g_i, \quad g_i \in N \tag{9}$$

Then, from (5) we have

$$(z)g_i = \{z + \alpha_{i1}, z + \alpha_{i2}, \ldots, z + \alpha_{im_i}\} \tag{10}$$

so $z^1 = z + \alpha_{i1}, \ldots, z^{mi} = z + \alpha_{im_i}$ are the nearest-neighboring points of z (which may include z itself) whose corresponding types <u>interact</u> with the type $(z)\tau$ at time t, that is, the present intensities of $(z^1)\tau, \ldots, (z^{m_i})\tau$ are the elements of the sequence of inputs $w_t \in \mathcal{D}_o(f^{\gamma_t})$ to $(z)\tau$ at time t.

Now we have all the ingredients for the description of the (local) dynamics of a DHA. $\mathcal{A} = \langle I, \tau \rangle$.

Let $v_t : I \to G$ be a family of functions indexed by the time variable $t \in T$, such that for any $z \in Z^k$, $(z)v_t$ is the structure label of the type assigned to z at time t. Similarly, let $w_t : I \to Q$ be a family of functions indexed by t such that for any $z \in Z^k$, $(z)w_t$ is the inten-

sity of the type assigned to z at time t.

The (local) dynamics of the DHA $\mathcal{A} = (I, \tau)$ is determined by specifying the structure and intensity transitions of the type in $\mathcal{A}$, these are given by

Structure transition at z:

$$(z)v_{t+1} = ((z)v_t, w_t | (z)v_t\Psi)H \tag{11}$$

Intensity transition at z:

$$(z)w_{t+1} = (w_t | (z)v_t\Psi)f^{(z)v_t} \tag{12}$$

with $f^{(z)v_t} \in \Phi \ \forall\ z \in Z^k$ and $\circ|\circ$ means "restricted to" in the set-theoretic sense. That is,

$$w_t | (z)v_t\Psi = \{(z + \alpha_{j1})w_t, (z + \alpha_{j2})w_t, \ldots, (z + \alpha_{jm_j})w_t\} \tag{13}$$

whenever

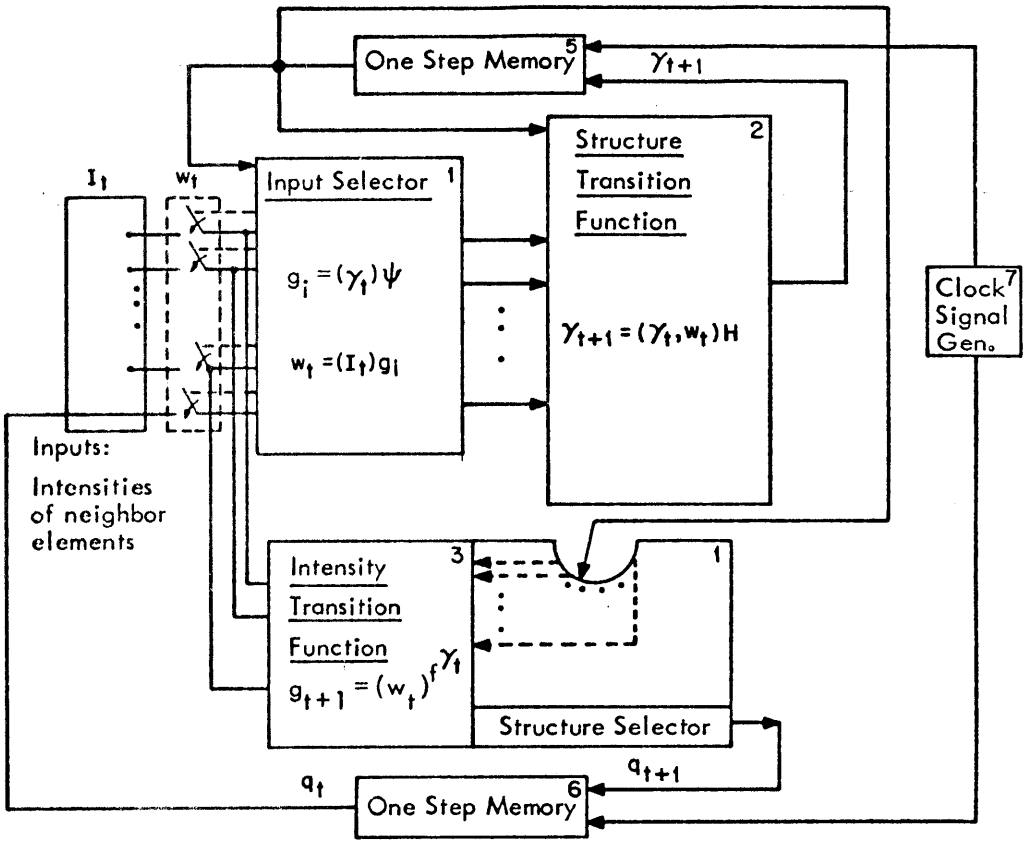
$$(z)v_t\Psi = (z)g_j \tag{14}$$

and

$$(z)g_j = \{z + \alpha_{ji}, z + \alpha_{j2}, \ldots, z + \alpha_{jm_j}\} \tag{15}$$

for some $g_j \in N$.

Figure 2 gives a graphical representation of the local dynamics of a DHA $\mathcal{A} = \langle I, \tau \rangle$. We note that this dynamics is determined at every $z \in Z^k$ by the state transition of $(z)\tau$, which is functionally dependent on the



Block Representation of a Type of Model

Figure 2

intensities of the types allocated at the nearest neighbor points of z.

In (12) is is understood that the expression is valid whenever the r.h.s. is defined (recall that, in general, $f^{(z)v_t} \in \Phi$ are partial functions), otherwise $(z)w_{t+1}$ is undefined.

Let $A = \langle I, \tau \rangle$ be a DHA. The state $(0,0) \in G \times Q$ of τ; that is the state corresponding to structure label 0 and intensity 0 is called the dormant state of τ and satisfies the following conditions.

$$(a) \quad n_o = 2k+1 \tag{16}$$

$$(b) \quad (0)\Psi = (z)g_o \tag{17}$$

where

$$\forall\, z \in Z^k \quad (z)g_o = \{z, z + \alpha_{0,1}, \ldots, z + \alpha_{0,2k+1}\}$$

and

$$\alpha_{0,i} = [0,\ldots,1,\ldots,0] \tag{18}$$

(1 in the ith position)

$$(c) \quad \forall\, w \in \mathcal{D}_o(f^o),\ w = q^1,\ldots,q^{2k+1}$$

$$(w)f^o = q^1 \tag{19}$$[1]

We note that (c) implies that the function f^o is total, i.e., $(w)f^o$ is defined for every $w \in \mathcal{D}_o f^o$.

---

[1] Note that the neighborhood function g_o orders the inputs of neighboring elements so that the state of the type itself is the first input.

(d) For any $\gamma \in G$, if $w = (0,\ldots,0) \in \mathcal{D}_o f^{\gamma}$ then

$$(w)f^{\gamma} = 0 \tag{20}$$

That is, if all the neighbors of τ determined by $\gamma\Psi$ have 0 intensities, the next intensity is zero independent of the structure.

(e) For $w_t = (0,\ldots,0)$, $\gamma_t = 0$,

$$(\gamma_t, w_t)H = 0 \tag{21}$$

The importance of the definition of dormant state relys on the fact that it allows us to characterize the information content of a type τ in A. Intuitively, an element in dormant state does not carry information.

Now, we establish a representation of the global dynamics of a DHA $A = \langle I,\tau \rangle$ in terms of the (local) dynamics of its type (given by (11)-(15)).

Let τ be the type of a DHA A, with intensity set Q and structure label G. A configuration C is a function that assigns to each point in the information space $I = Z^k$; a unique pair $(\gamma,q) \in G \times Q$ such that the support of C, $(C)\text{supp}_o$ is a finite set. $(C)\text{supp}_o$ is defined by

$$(C)\text{supp}_o = \{z \mid (z)C \neq (0,0)\} \tag{22}$$

In short, a configuration is an assignment of a structure and intensity to every point of the informational space I of a DHA A. Since at

every point of I a type τ is allocated we can express the configuration of A at any time t in terms of the structure label function v_t, and the intensity function w_t introduced proviously. Specifically,

$$C_t : I \to G \times Q \qquad \text{(t indicates time)}$$

can be represented by

$$(z)C_t = ((z)v_t, (z)w_t) \quad \forall\, z \in I \tag{23}$$

Clearly, since the $\{w_t\}$ are partial functions, C_t will be a partial function.

Expression (23) allows us to define a Configuration Transition Map (CTM for short), L in terms of the local dynamics of the corresponding type.

Let $A = \langle I,\tau \rangle$ be a given DHA. By C_A we denote the class of all configurations with finite support ($supp_o$) on I. The configuration transition map (CTM) L associated with A is the map

$$L : C_A \to C_A$$

such that given the configuration C_t of A at time t, C_tL is the configuration of A at time t+1.

The CTM L of $A = \langle I,\tau \rangle$ is expressed in terms of the local dynamics of A (i.e., the dynamics of τ) as follows:

Let V_A be the function space defined on I given by

$$V_A = (C_A)proj_1 \tag{24}$$

where $proj_1$ is the first coordinate projection function, i.e., for any

$$C_t = (v_t, w_t) \in C_A$$

$$(C_t)proj_1 = v_t \tag{25}$$

Similarly, let W_A be the function space on I given by

$$W_A = (C_A)proj_2 \tag{26}$$

where $proj_2$ gives the second coordinate of every $C_t = (v_y, w_t) \in C_A$:

$$w_t = (C_t)proj_2 \tag{27}$$

We see that C_A can be written as the cartesian product

$$C_A = V_A \times W_A \tag{28}$$

We now define the CTM L of $\dot{A}$ as the product of two maps; S and F

$$L = S \times F \tag{29}$$

where

$$S : V_A \times W_A \rightarrow V_A$$

is defined in terms of the local dynamics in $\dot{A}$ as,

$$(z)(v_t, w_t)S = ((z)v_t, w_t|(z)v_t\Psi)H \quad \forall\ z \in I \tag{30}$$

and

$$F : V_A \times W_A \to W_A, \text{ by}$$

$$(z)(v_t, w_t)F = (w_t | (z)v_t \Psi) f^{(z)v_t} \tag{31}$$

$$\forall z \in I \qquad f^{(z)v_t} \in \Phi$$

where $\circ|\circ$ means -restricted to- in the set theoretic sense.

Notice that at each time t, we have defined the configuration C_t of $A = \langle I, \tau \rangle$ as a partial function

$$C_t : I \to G \times Q$$

We note several things:

(1) The only portion of the domain of C_t that carries useful information is $(C_t)\text{supp}_o$.

(2) Although L maps entire configurations into entire configurations it is defined by <u>local</u> behavior ..., i.e., the calculation of $(z)(C\ L)$, for every $z \in I$, requires only the valuations of C at some or all the nearest neighboring points of z.

The last two observations have two important consequences:

(1) The finiteness of $(C_t)\text{supp}_o$, plus the finite amount of computation required for evaluating $(z)(C_t)L$ in terms of a finite set of nearest neighbors means that we can design efficient computational algorithms (the amount of computation is finite) for the simulation of the dynamics of A.

(2) More importantly, the local nature of L allows us to define several disjoint configurations (to be defined below) in A at any time t and propagate these <u>separately</u> until they interact (a concept that will be introduced shortly afterwards) at some future time t^1, ($t^1 > t$). This will be the cornerstone in our formulation of composite control processes in DHA's (see Section 4.5).

On the basis of the two concepts discussed above, we can express formally C_{t+1} as

$$C_{t+1} = (C_t | (C_t)\text{supp}_o)L \tag{32}$$

Two configurations C, C^1 in C_A are disjoint if

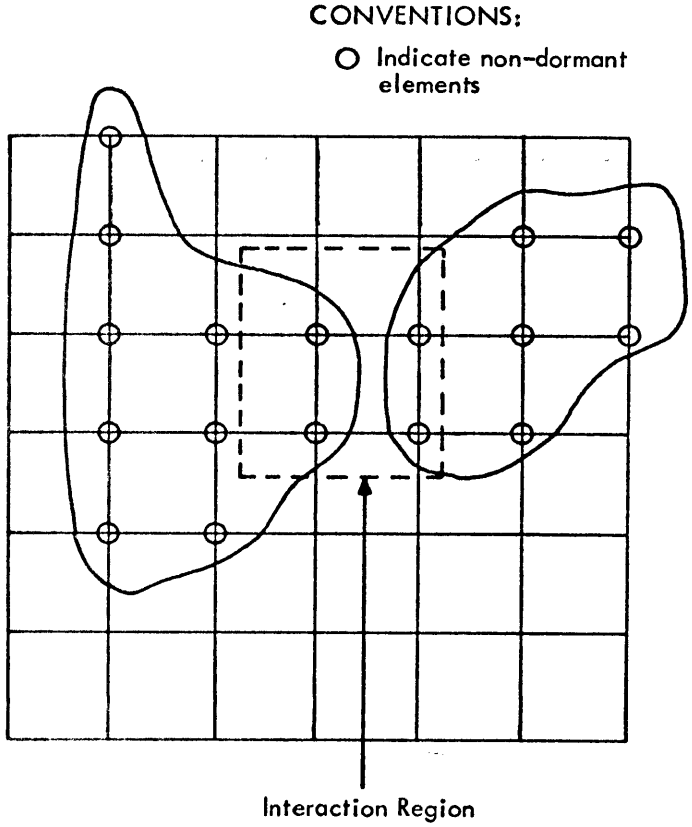
$$(C)\text{supp}_o \cap (C^1)\text{supp}_o = \emptyset \text{ (the empty set)} \tag{33}$$

where $\cap$ denotes set intersection.

Let C_t and C_t^1 be two disjoint configurations. We define the <u>disjoint union</u> $C_t \cup C_t^1$ of C_t and C_t^1 as the configuration

$$(z)C_t \cup C_t^1 = \begin{cases} (z)C_t & z \in (C_t)\text{supp}_o \\ (z)C_t^1 & z \in (C_t^1)\text{supp}_o \\ (0,0) & \text{otherwise} \end{cases} \tag{34}$$

It is clear that definition (34) can be inductively extended to any finite number of simultaneous disjoint configurations.



Two Interacting Configurations C_1, C_2 in Two-dimensional Information Space

Figure 3

Two disjoint configurations are said to be interacting (Figure 3) if at least one of the support points of one of them is in the nearest-neighborhood of one of the support points of the other.

Note that the non-interaction implies a bit more than nondisjointness. Specifically, let the closure $\overline{(C_t)\text{supp}_o}$ be defined intuitively as $C_t\text{supp}_o$ plus all those points in the nearest neighborhoods of points in $(C_t)\text{supp}_o$.

Mathematically, the closure $\overline{(C_t)\text{supp}_o}$ is given by the formula

$$(C_t)\text{supp}_o = (C_t)\text{supp}_o \cup \{z^1 \mid z^1 \in (z)g_j \text{ for some } z \in (C_t)\text{supp}_o,\ g_j = ((z)C_t\ \text{proj}_1)\Psi\} \tag{35}$$

then, interaction (resp. noninteraction) of C_t^1, C_t^2 is defined by

$$\overline{C_t^1\text{supp}_o} \cap \overline{C_t^2\text{supp}_o} \neq \emptyset \ (\text{resp. } = \emptyset) \tag{36}$$

We conclude this section with the definition of process trajectory, or simply process and some related important concepts.

Given a DHA $A = \langle I, \tau \rangle$, a process P in A is a sequence of configurations

$$P = \{C_o, C_1, \ldots, C_i, \ldots,\} \tag{37}$$

where $C_o \in C_A$

$$C_i = C_{i-1}L \qquad i=1,2,\ldots$$

We note that by the convention established above, P is really a sequence of supports of configurations. This allows us to have more than one process running simultaneously in A.

We say that two processes $P = \{C_o, C_1, \ldots\}$, $P' = \{C_o^1, C_1^1, \ldots\}$ do not interact if for each t, t=0,1,..., the configurations C_t and C_t^1 do not interact where t is time.

A subprocess $P_T^{T_o}$ of a process

$$P_T^{T_o} = \{C_{T_o}, C_{T_{o+1}}, \ldots, C_T\} \tag{38}$$

with $T_o \leq T$. This concept is frequently used in our study in cases in which we are interested only in particular subprocesses of a given process.

We now define a very important class of configurations in the study of DHA's. (This class is particularly important in the applications of DHA's studied in this thesis), the memory configurations. For this purpose we need the following concepts.

A configuration $C \in C_A$ is passive if the following condition holds:

$$(C)L\,|\,(C)\text{supp}_o = C\,|\,(C)\text{supp}_o \tag{39}$$

We note that this definition implies that the configuration can "expand" i.e., change its support in a time transition.[1]

[1]Although its values on $(C)\text{supp}_o$ remain fixed.

A configuration C is strongly passive if

(i) C is passive

and

(ii) $(z)C = (0, (z)w) \quad \forall\, z \in Z^k$.

That is, the configuration only carries information in the intensity part of the state and does not change with time provided that it does not interact with any other configuration; that is, its support is disjoint from the support of any non-passive configuration in I.

A memory configuartion $C \in C_A$ is defined as follows

(i) C is strongly passive

(ii) $(C \cup C')L|(C)\text{supp}_o = C|(C)\text{supp}_o \quad \forall\, C' \in C_A$ such that C' is disjoint from C. That is, the information contained in C is preserved in time.

Finally, let C_A be the set of configurations of $A = \langle I, \tau \rangle$, C any configuration in C_A and $z' \in I$ any point in the information space. We define the z'-translate of C, as the function $C' : I \to G \times Q$ given by

$$(z)C' = (z - z')C \quad \forall\, z \in I \tag{40}$$

where z - z' in the r.h.s. of (32) is perofrmed component wise.

It is clear that for any $z' \in I$, the z'-translate C' of $C \in C_A$ is also in C_A, i.e., its support is also finite.

A configuration C'_t is a subconfiguration of C_t if

$$\overline{(C'_t)\text{supp}_o} \subseteq \overline{(C_t)\text{supp}_o} \tag{41}$$

4.2 Basic Characteristics of DHA's

In this section we carry out an algebraic analysis of the model of DHA's introduced in Section 4.1.

The section is divided into two subsections: Subsection 4.2.1 is devoted to stating and examining some classes of DHA's with respect to local properties and to developing the tools for obtaining DHA's in these classes from a given DHA, such that the former are "equivalent" to the latter is a well-defined sense which is made explicit in each case. In this subsection we also introduce the rather important concept of covering DHA which will be extensively used in Section 4.4.

In Subsection 4.2.2 we introduce the concepts of behavior and display for DHA's. These concepts constitute a bridge between local and global characterizations of the dynamics of DHA's and are extensively used in Sections 4.3 and 4.4 in the construction of the minimal DHA of a lattice of DHA's with similar dynamics and in the construction of the covering DHA of a given set of DHA's which has the capabilities of simulating any of the DHA's in the set in a sense which is formulated in terms of their behaviors.

4.2.1 Local properties of DHA's

Throughout this subsection we assume as given a DHA $\mathcal{A} = \langle I, \tau \rangle$ where the type τ is characterized by a structure set Φ with corresponding (structure) label set G, intensity set Q, structure transition H, neighborhood functions set N, and input selector function Ψ, which were de-

fined in Section 4.1.

In virtue of the nearest-neighbor interaction scheme of A, is is sufficient for the analysis of dynamic local properties; that is, properties that depend on the one-step-in-time slate transition of τ, that we concentrate our efforts on a configuration U in $I = Z^k$, whose support consists of $\underline{0} = (0,\ldots 0) \in Z^k$ and its nearest neighboring points, i.e.,

$$(U)\text{supp}_o = \{z \in Z^k \mid (z, \underline{0})\ \rho \leq 1\} \qquad (1)$$

This is possible since each of the neighborhood function of $A = \langle I,\tau\rangle$ is invariant on I. Therefore, for any $z \in I$, we can translate the configuration C_t with support $(C_t)\text{supp}_o = \{z' \mid (z, z')\ \rho \leq 1\}$ by z units (i.e., $(z)\tau$ is translated to 0) without changing the information content of $(z)\tau$ at time t or its next state which is completely determined by the translation-invariant nearest neighboring types.[1]

Furthermore, the state transition at z given by

$$(z)v_{t+1} = (z)(v_t, w_t)S$$

$$(z)w_{t+1} = (z)(v_t, w_t)F$$

is functionally dependent on the intensities of the types allocated at the points

[1]This property in contradistinction with the characteristics of other types of distributed automata, on which, the location of the element in the information space is part of its information content. An example of this type of automata can be found in Wolpert [15].

$$(z)((z)v_t)^{\Psi} = (z)g_i = (z + \alpha_{i,1},\ldots,z + \alpha_{i,m_i})$$

for some $g_i \in N$. For $z = \underline{0}$ it is clear that $(z)g_i \subseteq (U)\text{supp}_o \; \forall \; g_i \in N$.

The first local property that we analyze in this section is state reachability. Before defining this property, we introduce three concepts which are the ingredients of the definition; these are:

1. Initial structure label set I_G: I_G is a subset of the structure label set G satisfying the following conditions: I_G is non-empty and at least one of its elements is different from the dormant label 0.

2. Initial intensity set I_Q: I_Q is a subset of the intensity set satisfying the following conditions: I_Q is non-empty and at least one of its elements is different from the dormant intensity 0.

3. Initial configuration set iCS_A: iCS_A is a finite subset of configurations C such that $\forall \; C \in iCS_A$

$$(z)C \in I_G \times I_Q \quad \forall \; z \in (C)\text{supp}_o \tag{2}$$

A state $(\gamma,q) \in G \times Q$ of τ is reachable if there exists some $C_o \in iCS_A$ and $\sigma < \infty$ such that the sequence of configurations $\{C_t | t=0,\ldots\sigma\}$ given by

$$C_oL = C_1,\ C_1L = C_2,\ldots,\ C_{\sigma-1}L = C_\sigma$$

satisfies (3)

$$(z)C_\sigma = (\gamma,q) \text{ for some } z \in I.$$

where L is the CTM of A.

The reachable structure label G_r of A is a subset of G given by

$$G_r = \{\gamma \in G | (\gamma,q) \text{ is reachable for some } q \in Q\} \qquad (4)$$

Similarly, the reachable intensity set Q_r of A is a subset of Q given by

$$Q_r = \{q \in Q | (\gamma,q) \text{ is reachable for some } \gamma \in G\} \qquad (5)$$

The reachable state set R of T is a subset of the cartesian product $G_r \times Q_r$ given by

$$R = \{(\gamma,q) | \gamma \in G_r,\ q \in Q_r,\ (\gamma,q) \text{ is reachable}\} \qquad (6)$$

We say that A is reachable if

$$G = G_r$$

$$Q = Q_r$$

Note that this does <u>not</u> mean that every pair (γ,q) is reachable and hence our reachable DHA need not be minimal in the sense one usually thinks of for finite-state machines. We are willing to sacrifice this property in favor of our weaker notion, since our two-level structure-intensity state description is both appealing on physical grounds and useful for several mathematical manipulations we will perform later.

The second characteristic of A that we are concerned with is <u>com-</u>

pleteness. We state it in terms of configurations on U.

We say that the structure label $\gamma \in G$ is complete if the corresponding structure

$$f^{\gamma} : \mathcal{D}_o f^{\gamma} \to Q$$

is a total function; that is, $(w)f^{\gamma}$ is defined for every $w \in \mathcal{D}_o f^{\gamma}$. We say that A is complete if every structure label $\gamma \in G$ is complete.

We recall from our definition of the structure-set Φ in Section 4.1, that the only structure label which is guaranteed to be complete is the 0 (zero) label, corresponding to the dormant structure f^o.

We say that A is deterministic if $(0)\tau$ in U satisfies the following conditions:

(i) The set $I_G \times I_Q$ is a singleton say $\{(\gamma_o, q_o)\}$ with $\gamma_o \neq 0$ and/or $q_o \neq 0$.

(ii) A is complete.

Most of the remaining part of this subsection is devoted to answering the following questions:

a) Given a DHA A, does there exist a complete DHA A_C such that its dynamic behavior is equivalent to that of A in a well-defined sense?

b) Given a complete DHA A does there exist a deterministic DHA A_D whose dynamical behavior is equivalent to that of A?

c) Given a complete DHA A (not necessarily deterministic) does there exist a reachable DHA A_r such that its dynamic behavior

is equivalent to that of A?

We will show that the answer to questions a), b), and c) above is affirmative for the three cases.

These answers are provided by means of corresponding computational algorithms for A_C, A_D and A_r, given A.

Further we will show that the algorithm for computing A_r, can be utilized, with some minor modifications, for computing a DHA A_T called the trim DHA associated with A.

Finally, we will discuss two important properties of the DHA's A_C, A_D, A_r, and A_T associated with a given DHA A and some constraints on C_A, called functionality and conflict free conditions.

In order to give answers to questions a), b) and c) above, we need several concepts relating the local dynamics of two DHA's. The first of these is the concept of covering DHA $A' = \langle I', \tau' \rangle$ with initial state set $I'_G \times I'_Q$ of a given DHA $A = \langle I, \tau \rangle$ with initial state set $I_G \times I_Q$.

We say that A' covers A if the following holds:

(i) $I' = Z^{k'}$, $I = Z^k$ and $k' \geq k$ (7)

(ii) There exist partial morphisms[1] ψ_1, ψ_2

[1]Partial morphisms are partial functions which, when defined, have the properties of ordinary morphisms between similar algebraic structures. In Appendix A.1 we discuss partial morphisms in the context of universal (partial) algebras of the type that appears in our model of DHA's. For a general discussion of partial morphisms the reader is referred to Gratzer [ ].

$$\psi_1 : G^1 \text{ onto } G$$

$$\psi_2 : Q^1 \xrightarrow{\text{onto}} Q$$

that possess the following properties:

$$1 - \quad I_G = I_{G^1\psi_1} , \quad I_Q = I_{Q^1\psi_2} \tag{8}$$

2 - Let H, H' be the structure transition functions of τ and τ' respectively; then for any $\gamma' \in G'$ and any $\omega' \in \mathcal{D}_o(f^{\gamma'}) \subseteq (\hat{Q}')^{2k'+1}$

$$(\gamma'\psi_1, \omega'\hat{\psi}_2)H \subseteq (\gamma', \omega')H'\psi_1 \tag{9}$$

$$(\omega'\hat{\psi}_2)\, f^{\gamma'\psi_1} \subseteq (\omega')\, f^{\gamma'}\psi_2 \tag{10}$$

$$f^{\gamma'\psi_1} \in \Phi' \text{ of } A'$$

where $\hat{\psi}_2$ in (9) and (10) is the unique extension of ψ_2 from $Q' \to Q$ to $(\hat{Q}')^{2k'+1} \to (\hat{Q})^{2k+1}$ defined by

For any $\omega' = (q_o, \ldots, q_m) \in (\hat{Q}')^{2k'+1}$

$$\omega'\hat{\psi}_2 = \begin{cases} (q_o\psi_2, \ldots, q_m\psi_2) \in (\hat{Q}^{2k+1})^1 & \text{if } m \leq 2k+1 \\ \emptyset & \text{if } m > 2k+1 \end{cases} \tag{11}^1$$

---

[1] Notice that in , $|\omega'| = |\omega'\hat{\psi}_2|$ when $\omega'\hat{\psi}_2$ is defined. Partial morphisms with this property are called fine partial morphisms (see Eilenberg [2]).

The inclusion operations ($\underline{\subset}$) in (9) and (10) are to be understood in the following sense:

If the left-hand side is defined, then (9) becomes

$$(\gamma'\psi_1, \omega'\hat{\psi}_2)H = (\gamma', \omega')H'\psi_1 \tag{12}$$

If the left-hand side of (10) is defined then

$$(\omega'\hat{\psi}_2)f^{\gamma'\psi_1} = (\omega')f^{\gamma'}\psi_2 \tag{13}$$

If the left-hand side of (9) (resp.(10)) is not defined we assign to $(\gamma'\psi_1, \omega'\hat{\psi}_2)H$ (resp. $\omega'\hat{\psi}_2 f^{\gamma'\psi_1}$) the $\emptyset$ valuation ($\emptyset$, the empty set).

In synthesis (9) means (12) and (10) means (13) if the corresponding left-hand sides are defined, and if either of them is not defined we equate it to $\emptyset$, which will be always included in the corresponding right-hand side even if it is not defined. However, whenever either left-hand side of (6) or (7) is defined, we demand the corresponding right-hand side to be defined.

We note that (12) holds only if $\gamma'\psi_1$, $\omega'\hat{\psi}_2$ are defined. Similarly, (13) holds only if $\gamma'\psi_1$ is defined, $(\omega')f^{\gamma'}\psi_2$ is defined and $(\omega')\hat{\psi}_2$ is defined. But $(\omega')f^{\gamma'}\psi_2$ is defined only if $(\omega')f^{\gamma'}$ is defined and this requires that $|\omega'| = n\gamma'$. Also $(\omega'\hat{\psi}_2)f^{\gamma'\psi_1}$ is defined only if $\omega'\hat{\psi}_2$ is defined and $|\omega'\psi_2| = n_{\gamma',\psi_1}$ thus we conclude that (9) and (10) hold only if

$$n_{\gamma'} = n_{\gamma',\psi_1} \tag{14}$$

We summarize the observations above in the

Lemma 1. A necessary condition for (12) and (13) to hold is,

For $\gamma' \in G'$ such that $\gamma'\psi_1$ exists, $n_{\gamma'} = n_{\gamma'\psi_1}$ and $|\omega' \hat{\psi}_2| = n_{\gamma'}$. ▲

(iii) Let N, N' be the sets of neighborhood functions of A, A' and Ψ, Ψ' their respective input selector functions.

There exists an onto (total, see Gratzner [ ]) morphism $\psi_3 : N' \to N$ such that if $g'_j \in N$ $g'_j : I \to 2^I$, $j \leq 2^{2k'}$, $g'_j\psi_3 = g_i$ $g_i : I \to 2^I$ $i \leq 2^{2k}$ then the characteristics m_j of g'_j and m_i of g_i satisfy,

$$m_j = m_i \tag{15}$$

For any $z \in I$ $z' \in I'$

$$(z)g_i = \{z + \alpha_{i,1}, \ldots, z + \alpha_{i,m_i}\} \in 2^I$$

$$(z')g'_j = \{z' + \alpha'_{j1}, \ldots, z' + \alpha_{j,m_j}\} \in 2^I$$

we demand

$$(\alpha_{i,k}, \tilde{0}) = \alpha'_{jk} \quad k=1,\ldots,m_i \tag{16}$$

where $\tilde{0} = (0,0\ldots0) \in Z^{k'-k}$.

Condition (15) guarantees that $(z')g'_j$ and $(z)(g'_j\psi_3)$ have the same number of nearest-neighboring points in I' and I, respectively. Condition (16) guarantees that the geometry of neighborhoods of points in I' are preserved for the corresponding points in I. This is illustrated

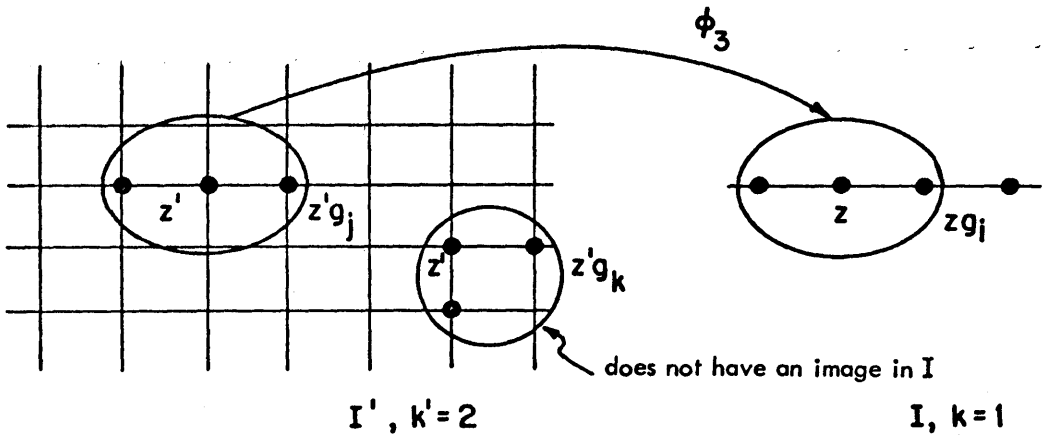


Illustration of the Characteristics of the Morphism ϕ_3 in the covering of a DHA $A' = \langle Z^2, \tau' \rangle$ with a DHA $A = \langle Z^1, \tau \rangle$

Figure 1

in Figure 1.

In short $g_i = g'_{j\,3}$ if g'_j can be embedded in I, as illustrated in Figure 1.

(iv) Finally, we require the following compatibility condition in terms of the morphisms ψ_1 defined in (ii), ψ_3 defined in (iii) and the input selector functions Ψ, Ψ' of $\acute{A}$ and $\acute{A}'$:

$$\psi_1 \Psi = {}^{\Psi'}\psi_3 \tag{17}$$

i.e., for any $\gamma' \in G'$, if $\gamma'\psi_1 = \gamma \in G$, and $\gamma\Psi = g_i \in N$, then we demand $\gamma'\Psi' = g'_j \in N'$ and $g'_j{}^{\Psi}{}_3 = g_i$.

We summarize the definition and properties of the covering DHA $\acute{A}'$ of a given DHA $\acute{A}$ in the commutative diagrams of Figure 2.

Notice that if

$$(\gamma_o, q_o), \ldots, (\gamma_t, q_t), \ldots$$

is a sequence of states assumed by $(\underline{0})\tau$ of $\acute{A}$, with

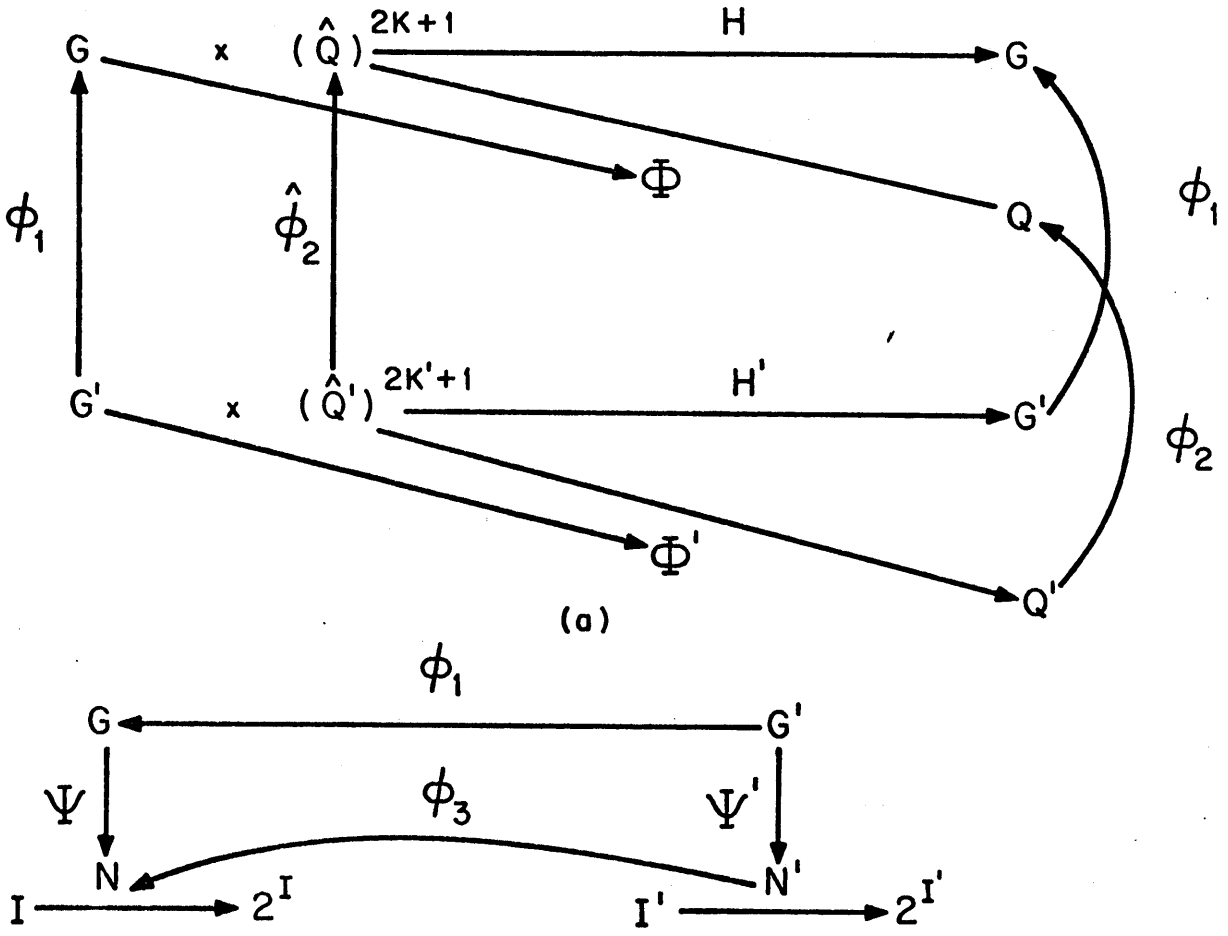
$$\gamma_{\sigma+1} = (\gamma_\sigma, \omega_\sigma | (\gamma_\sigma)\Psi)H$$

$$(\underline{0})\omega_{\sigma+1} = q_{\sigma+1} = (\omega_\sigma | \gamma_\sigma \Psi) f^{\gamma_\sigma} \qquad \sigma = 0, \ldots, t$$

$$\omega_o | \gamma_o \Psi \in \mathrm{ICS} \subseteq \hat{I}_Q \cap \mathcal{D}_o f^{\gamma_o}, \quad \gamma_o \in I_G$$

and $\omega_\sigma : U \to Q$

Then, the definition of the covering DHA $\acute{A}'$ given above guarantees



Cummutative Diagrams Representing the covering of DHA $A = \langle I, \tau \rangle$ by DHA $A' = \langle I', \tau' \rangle$.
(a) dynamic interrelation between types τ and τ'
(b) neighborhood interrelation in I and I'

Figure 2

guarantees that there exists a sequence of states assumed by $(\underline{0})\tau'$[1] of A'

$$(\gamma'_o,q'_o),\ldots,(\gamma'_t,q'_t),\ldots$$

with

$$\gamma'_{\sigma+1} = (\gamma'_\sigma,\ \omega'_\sigma|(\gamma'_\sigma)\Psi')H'$$

$$q'_{\sigma+1} = (\omega'_\sigma|(\gamma'_\sigma)\Psi')f^{\gamma'_\sigma} \qquad \sigma=0,\ldots,t$$

and

$$\omega'_\sigma : U' \to (\hat{Q}')^{2k+1} \qquad (\gamma'_o,q'_o) \in I'_G \times I'_Q$$

that satisfies

$$\gamma_\sigma = \gamma'_\sigma\psi_1$$

$$q_\sigma = q'_\sigma\psi_2 \qquad \sigma=0,\ldots,t$$

$$\hat{\omega}_\sigma = \omega'_\sigma\hat{\psi}_2$$

$$\gamma_\sigma\Psi = \gamma'_\sigma\Psi'\psi_3$$

This follows from (ii)-1 and (ii)-2 above.

We say that two DHA's, A, A' are (partially) <u>isomorphic</u> if A covers A' and A' covers A. We note that (i) implies

---

[1]We note that $\underline{0}$ in $(\underline{0})\tau'$ is the origin in $Z^{k'}$ while $\underline{0}$ in $(\underline{0})\tau$ is the origin in Z^k; we use the same symbol for both since there is no risk of confusion.

$$I_G \times I_Q = I'_G \times I'_Q \qquad (15)$$

and from expressions (9), (10), ψ_i is an isomorphism for i=1,2.

Remark: The definition of covering DHA given in this subsection is the basis of the construction of a covering DHA of a family of DHA's, a task that will be carried out in Section 4.4.

Now we proceed to give an answet to question a). This is provided by

Proposition 1. Let $A = \langle I, \tau \rangle$ be a given DHA, with initial state set $I_G \times I_Q$. Then, there exists a complete DHA $A' = \langle I', \tau' \rangle$ with initial state set $I_G' \times I_Q'$ such that A' covers A.

Proof:

Let I' = I (i.e. k=k') and define τ' as follows:

$$I_G' = I_G$$

$$I_Q' = I_Q$$

$$G' = G$$

$$H' = H$$

Let $Q = \{0,\ldots,R\}$, R and integer, be the intensity set of τ and define Q', the intensity set of τ' as

$$Q' = Q \cup \{R+1\} \qquad R+1 \notin Q$$

The set Φ' of structures τ' is constructed as follows:

For each $f^\gamma \in \Phi$, (the set of structures of τ) $f^\gamma : \mathcal{D}_o f^\gamma \to Q$, define $\tilde{f}^\gamma \in \Phi'$ as follows:

$$\tilde{f}^{\gamma} : \mathcal{D}_o \tilde{f}^{\gamma} \to Q'$$

where

$$\mathcal{D}_o \tilde{f}^{\gamma} = \{\omega' \in (\hat{Q}')^{2k+1} : |\omega'| = n_{\gamma}\} \tag{16}$$

and

$$(\omega')\tilde{f}^{\gamma} = \begin{cases} (\omega')f \text{ if } (\omega')f^{\gamma} \text{ is defined} \\ \\ R+1 \text{ if } (\omega')f^{\gamma} \text{ is not defined or } \omega' \in \mathcal{D}_o \tilde{f}^{\gamma} - \mathcal{D}_o f^{\gamma} \end{cases} \tag{17}$$

Notice that in the second line in the r.h.s. of (17) the condition $\omega' \notin \mathcal{D}_o f^{\gamma}$ means that at least one entry in the sequence ω' equals R+1. Notice also that $\tilde{f}^{\gamma}$ $\forall$ $\gamma \in G'$ is a complete function on $\mathcal{D}_o \tilde{f}^{\gamma}$. Finally, let

$$N' = N$$

$$\Psi' = \Psi$$

where N, N' are the neighborhood functions of $\mathcal{A}$ and $\mathcal{A}'$ and Ψ', Ψ their corresponding input selector functions.

In order to show that $\mathcal{A}'$ covers $\mathcal{A}$ we need to exhibit three (3) morphisms ψ_1, ψ_2, ψ_3 and show that they satisfy the properties described in (i)-(iv) above. Let

$\psi_1 : G' \to G$ be the identity

$\psi_2 : Q' \to Q$ defined by

$$q'\psi_2 = \begin{cases} q' \text{ if } q' \in Q \\ \\ \emptyset \text{ if } q' \underline{\in} \{R+1\} \end{cases}$$

and

$$\psi_3 : N' \to N \text{ the identity.}$$

Clearly, with these definitions, conditions (i)-(iv) above are satisfied and A' covers A.

Corollary. The subset ∇ of $(\hat{Q}')^{2k'+1}$ defined by

$$\nabla = \{\omega' \in (Q')^{2k+1} : \text{At least one entry of } \omega' \text{ is equal to } R+1 \tag{18}$$

is an algebra ideal of the algebra $\langle Q', \Phi \rangle$ [1]

Proof: Follows at once from (17). ▲

The importance of the ideal ∇ defined in the corollary above will become apparent later.

Now we consider question b).

Proposition 2. Let $A = \langle I, \tau \rangle$ be a complete DHA with initial state set $I_G \times I_Q$. There exists a deterministic DHA $A_D = \langle I, \tau_D \rangle$ with initial state set $\{(\tilde{\gamma}_o, \tilde{q}_o)\}$ and a family M of pairs of morphisms,

$$M = \{(h_i^1, h_i^2), i=1,\ldots,\ell = |I_G \times I_Q|\}, \; h_i^1 : G_D \to G, \; h_i^2 : Q^D \to Q$$

$i=1,\ldots,\ell$, where G_D and Q_D are the structure and intensity sets of A_D, such that

$$I_G = \{\tilde{\gamma}_o h, \ldots, \tilde{\gamma}_o h_\ell^1\}$$

[1] See sppendix A.1 for definition of $\langle Q', \Phi \rangle$ and algebra ideals.

$$I_Q = \{\tilde{q}_o h_i^2, \ldots, \tilde{q}_o h_\ell^2\} ,$$

and for every sequence of states

$$(\gamma_o, q_o), \ldots, (\gamma_t, q_t), \ldots, \text{ with } (\gamma_o, q_o) \in I_G \times I_Q$$

taken by some $(z)\tau$, there exists a sequence of states taken by $(z)\tau_D$

$$(\tilde{\gamma}_o, \tilde{q}_o), \ldots, (\tilde{\gamma}_t, \tilde{q}_t) \ldots; (\tilde{\gamma}_t, \tilde{q}_t) \in G_D \times Q_D \quad t=0, \ldots,$$

satisfying

$$\left.\begin{array}{l} \tilde{\gamma}_t h_i^1 = \gamma_t \quad \tilde{q}_t \in h_i^2 = q_t \quad t=0,1,\ldots \text{ for exactly one} \\ \text{pair } (h_i^1, h_i^2) \text{ in M.} \end{array}\right\} \quad (19)$$

Proof:

We first construct A_D and then show that it satisfies condition (19). Define G_D and Q_D, as follows:

$$\begin{aligned} G_D &= (\tilde{G})\ E_\ell \\ Q_D &= (\tilde{Q})\ E_\ell \end{aligned} \quad (20)$$

where $\tilde{G}$ is the set of all ℓ-tuples of elements of G, $\tilde{Q}$ is the set of all ℓ-tuples of elements of Q and E_ℓ is the ℓth-iterate of the cantor pairing function E_2 (see Davis [ ]).

$$E_2 : IN^2 \to IN \text{ , IN the set of non-negative integers,}$$

$$(n_1,n_2)E_2 = \frac{1}{2}\left((n_1 + n_2)^2 + 3n_1 + n_2\right) \tag{21}$$

the S-th iterate of E_2,

$$E_S : IN^S \to IN \qquad S > 2$$

defined inductively by

$$(n_1,\ldots,n_S)E_S = ((n_1,\ldots,n_{S-1})E_{S-1},\ n_S)E_2 \tag{22}$$

It is easy to prove (see Yasuhara [ ]), that the cantor function and every one of its iterates are 1-1 onto functions.

Let $\hat{E}_\ell : \hat{\tilde{Q}}^{2k+1} \to \hat{Q}_D^{2k+1}$ be the extension component-wise of E_ℓ from $\tilde{Q}$ to the set of sequences of elements of $\tilde{Q}$ (i.e., $\hat{\tilde{Q}}^{2k+1}$) of length at most 2k+1.

Let

$$\begin{aligned} &\hat{e}_i^\ell : \hat{\tilde{Q}}^{2k+1} \to \hat{Q}^{2k+1} \\ &\qquad\qquad\qquad\qquad i=1,\ldots,\ell \\ &e_i^\ell : \tilde{G} \to G \end{aligned} \tag{23}$$

be the ith projection functions on $\hat{\tilde{Q}}^{2k+1}$ and $\tilde{G}$ respectively.

Now we construct the set of structures Φ^D of A^D. For each $\tilde{\gamma} \in G_D$ define

$$f_D^{\tilde{\gamma}} : \mathcal{D}_o f^{\tilde{\gamma}} \to Q_D$$

where

$$\mathcal{D}_o f^{\tilde{\gamma}} = \{\tilde{\omega} \in \hat{Q}_D^{2k+1} \mid |\tilde{\omega}\ \hat{E}_\ell^{-1}\ \hat{e}_i^\ell| = n_{\tilde{\gamma} E_\ell^{-1} e_i^\ell}\ i=1,\ldots,\ell\} \tag{24}$$

For each $\tilde{\omega} \in \mathcal{D}_o f^{\tilde{\gamma}}$

$$(\tilde{\omega}) f^{\tilde{\gamma}} = (\tilde{\omega} E_\ell^{-1} \hat{e}_1^\ell f^{\gamma_1}, \ldots, \tilde{\omega} E_\ell^{-1} \hat{e}_\ell^\ell f^{\gamma_\ell}) E_\ell \tag{25}$$

where $\tilde{\gamma} = (\gamma_1, \ldots, \gamma_\ell) E_\ell$.

The structure transition function H_D of A_D is defined by

$$H_D : G_D \times \hat{Q}_D^{2k+1} \to G_D$$

for $\tilde{\gamma} \in G_D \quad \tilde{\omega} \in \hat{Q}_D^{2k+1}$

$$(\tilde{\gamma}, \tilde{\omega}) H_D = ((\gamma_1, \tilde{\omega} \hat{E}_\ell^{-1} \hat{e}_1^\ell) H, \ldots, (\gamma_\ell, \tilde{\omega} \hat{E}_\ell^{-1} \hat{e}_\ell^\ell) H) E_\ell \tag{26}$$

The set of neighborhood functions N_D is defined by,

$$N_D = \hat{N} \tag{27}$$

where $\hat{N}$ is the set of all ℓ-tuples of elements of N. The input selector function Ψ_D of A_D is given by

$$\forall\ \tilde{\gamma} \in G_D$$

$$\tilde{\gamma} \Psi_d = \bigcup_{\substack{\gamma_i \in \tilde{\gamma} E_\ell^{-1} (\gamma_i) \Psi \\ i=1,\ldots,\ell}} \tag{28}$$

Finally, the initial structure label $\tilde{\gamma}_o$ and initial intensity $\tilde{q}_o$ of A_D are given by

$$\tilde{\gamma}_o = (I_G) E_\ell \ , \ \tilde{q}_o = (I_Q) E_\ell \tag{29}$$

To show that A_D as constructed above satisfies (19), we must construct the set of morphisms M.

Let $I_G \times I_Q = \{(\gamma^o, q_1^o), \ldots, (\gamma_i^o, q_i), \ldots, \gamma_\ell^o, q_\ell^o\}$ with the natural ordering of the integers on the subindex i, i=1,... , .

Define

$$h_i^1 = E_\ell^{-1} e_i^\ell$$

$$i=1,\ldots,\ell \qquad (30)$$

$$h_i^2 = E_\ell^{-1} e_i^\ell$$

Since there is one pair (h_1^i, h_2^i) for each initial state, it is clear that $I_G \times I_Q \to M$ is onto; also since E_ℓ is a bijection, this map is 1-1. Condition (19) follows at once from (24)-(26).

Remark: The construction of A_D given in the proof of Proposition 2, involves, from the operational point of view, running at each step transition (in time) ℓ step transitions of A, one for each of its ℓ-possible initial states. Based on this construction, we could develop an algorithm for the construction of A_D, given A and its initial state set. However, deterministic DHA is used in this study only for theoretical purposes (i.e., proving additional results) and therefore we do not need such an algorithm.

The deterministic DHA A_D associated with a given complete DHA A, as constructed in the proof of the proposition, is said to emulate A.

At this point, before passing to consider question c) we consider the following obvious facts.

Proposition 3.

(i) Let A, A', A'' be DHA's such that A' covers A, and A'' covers A'; then A'' covers A.

(ii) Let A be a not necessarily complete DHA. Then, there exists a DHA A_D that emulates the complete DHA covering A.

Now we consider question c).

Proposition 4. Let $A = \langle I, \tau \rangle$ be a complete DHA with initial state set $I_G \times I_Q$ and initial configuration set ics_A. Then there exists a reachable DHA $A^r = \langle I, \tau^r \rangle$ with initial state set $I_G^r \times I_Q^r$, that emulates A.

Proof:

The proof is constructive:

set

$$ics_{A_r} = ics_A \tag{31}$$

We first construct the reachable set R of A. Let $\theta_o = ics_A$. (Recall that by definition ics_A is finite.) Let $\{\theta_i\}$, be the sequence of sets of configurations defined by

$$\theta_{i+1} = \theta_i L - \bigcup_{j=o}^{i} \theta_j \qquad i=0,1,\ldots . \tag{32}$$

Let A_i be the following sequence of subsets of $G \times Q$.

$$A_{i+1} = \{(\gamma,q) \in G \times Q \mid (\gamma,q) = (z)C \text{ for some } z \in I,\ C \in \theta_{i+1}\} - \bigcup_{j=o}^{i} A_j \tag{33}$$

with

$$A_o = \{(\gamma,q) \in G \times Q \mid (\gamma,q) = (z)C,\ \text{some } z \in I,\ C \in \text{ics}_A\} \tag{34}$$

Clearly θ_i, A_i i=0,1, are finite sets. The sequence $\{\theta_i\}$ consists of mutually disjoint subsets of configurations with finite support. Similarly, the sequence $\{A_i\}$ consists of subsets of $G \times Q$ which are mutually disjoint.

From the observations above, we conclude that if for some $\ell>0$ $\theta_\ell = \emptyset$ (the empty set), then $\theta_{\ell+s} = \emptyset$ $\forall$ $s>0$ and thus from (33) we have that $A_{\ell+s} = \emptyset$.

The reachable set R of $\mathcal{A}$ is then given by

$$R = \bigcup_{j=o}^{m} A_j \tag{35}$$

where

$$m = \max j \text{ such that } A_j \neq \emptyset \tag{36}$$

It is clear that m is a finite positive integer since R is a finite set. Unfortunately, due to the unboundedness of the informational space, no effective procedure can be given for estimating the value of m (i.e., no effective halting condition exists for the computation described by

(35)). Therefore, the procedure based on (35) and (36) is not an algorithm. Let

$$I_{G_r} = \{\gamma \in G \mid \gamma = (z)Cproj_1 \text{ for some } z \in I,\ C \in ics_A\}$$
$$I_{Q_r} = \{q \in Q \mid q = (z)Cproj_2 \text{ for some } z \in I,\ C \in ics_A\} \tag{37}$$

and

$$G_r = Rproj_1$$
$$Q_r = Rproj_2 \tag{38}$$

The set of structures Φ_r of A_r is given by

$$\Phi_r = \{f^\gamma \in \Phi \mid \gamma \in G_r\} \tag{39}$$

The structure transition function H_r of A_r is defined by

$$H_r = H \mid (G_r \times \hat{Q}_r^{2k+1}) \tag{40}$$

The set of neighborhood functions N_r is given by

$$N_r = G_r \Psi \tag{41}$$

and the input selector function Ψ_r is given by

$$\Psi_r = \Psi \mid G_r \tag{42}$$

A_r as constructed above, emulates A since by (31) $ics_{A_r} = ics_A$ and by (38) G_r and Q_r contain structures and intensities that are reachable. ▲

Remark: Next we will introduce a new type of DHA whose definition is more restrictive than that of reachable DHA's and for which an effective procedure (algorithm) for estimating m can be devised; this DHA is called the trim DHA. In Appendix A.2 we give an algorithm for computing such DHA and discuss some aspects related to the algorithm's computational complexity and its implementability as a parallel procedure.

Trim DHA's play an important role in the application we are considering in this study.

Let $A = \langle I, \tau \rangle$ be a DHA, with initial state set $I_G \times I_Q$ and initial configuration set ics_A. Let T_H be a positive integer that we call the horizon of A.

We say that the state $(\gamma, q) \in G \times Q$ of τ, is trim if there exists an integer $\sigma \leq T_H$ and a sequence $\{C_t, t \leq T_H\}$ of configurations on I such that

$$\left.\begin{aligned} & C_o \in ics \\ & (z)C_{t+1} = (z)(C_t L) \qquad \forall\, z \in U, \quad t \leq \sigma - 1 \\ \text{and} \quad & (z)C_\sigma = (\gamma, q) \text{ for some } z \in U. \end{aligned}\right\} \tag{43}$$

The subset of ∇ of $G \times Q$ composed of those states of $(z)\tau, z \in U$, that are trim is referred to as the trim subset of A.

The trim DHA A_T associated with a given DHA $A = \langle I, \tau \rangle$ is defined

as follows:

$$ics_{A_T} = ics_A \tag{44}$$

Initial label and intensity sets:

$$I_{G_T} = (\nabla)proj_1 \subseteq I_G$$

$$I_{Q_T} = (\nabla)proj_2 \subseteq I_Q$$

Label and intensity sets: (45)

$$G_T = (\nabla)proj_1$$

$$Q_T = (\nabla)proj_2$$

Structure transition function:

$$\forall\ \gamma \in G_T \quad \omega \in \hat{Q}_T^{2k+1}$$

$$(\gamma, \omega)H_T = \begin{cases} (\gamma,\omega)H \text{ if } (\gamma,\omega)H \in G_T \\ 0 \quad \text{otherwise} \end{cases} \tag{46}$$

Structure set Φ_T

$$\Phi_T = \{\tilde{f}^{\gamma} | \gamma \in G_T \subset G\}$$

$$\forall\ \omega \in \mathcal{D}_o\tilde{f}^{\gamma},$$

$$(\omega)\tilde{f}^{\gamma} = \begin{cases} (\omega)f^{\gamma} \text{ if } (\omega)f^{\gamma} \in Q_T \\ 0 \quad \text{otherwise} \end{cases} \tag{47}$$

where

$$\mathcal{D}_o\tilde{f}^{\gamma} = \{\omega \in \mathcal{D}_o f^{\gamma} \,|\, |\omega| = n_{\gamma}\} - \{\omega \in \mathcal{D}_o f^{\gamma} - \hat{Q}_T^{2k+1}\}$$

Neighborhhod function set N_T:

$$N_T = (G_T)^{\Psi} \tag{48}$$

Input selector function Ψ_T:

$$\Psi_T = \Psi|G_T \tag{49}$$

In the proof of proposition 4 let

$$\theta_o = Ics \tag{50}$$

and

$$m = \min (T_H, \{maxi, \text{ such that } \theta_i \neq \emptyset\}) \leq T_H \tag{51}$$

In this case, since m is bounded by T_H the procedure ((32)-(34)) for computing the trim DHA associated with a given DHA $\mathcal{A}$ is an algorithm. Its implementation is given in Appendix A.2.

Notice that

$$|\nabla| \leq R \leq |G \times Q| \tag{52}$$

We conclude this subsection with two important related aspects of the dynamics of DHA's. These are <u>functionality</u> and <u>conflict-free</u> conditions, (see Gallagher [ ] for discussion of these concepts for general recursions with side constraints).

Given a trim DHA $\mathcal{A} = \langle I, \tau \rangle$ with ics = A and horizon T_H and $C \in C_A$ a configuration, we say that $\mathcal{A}$ is <u>functional with respect to C</u> if there exists a configuration $C_o \in A$ and an integer $\sigma \leq T_H$ such that

$$C_o L^{\sigma} = C$$

We say that A is conflict-free with respect to a set M of configurations of C_A if it is functional with respect to each $C \in M$.

Finally, A is sequentially conflict-free if M is a process P^T with $T \leq T_H$ in A.

4.2.2 The concepts of behavior and display in DHA's

Let $A = \langle I, \tau \rangle$ be a complete trim DHA with initial state set $I_G \times I_Q$, and CTM L. Let C_A be the class of cinfigurations of A.

We define the σ=iterate of L, L^{σ}, $\sigma \geq 0$ an integer as the map

$$L^{\sigma} : C_A \to C_A$$

where L^{σ} is the σ-fold-composition of L with itself, (L^{o} = identity).

Since A is trim and complete L^{σ} is well defined for every $C \in C_A$ such that (z) $C \in I_G \times I_Q \quad \forall z \in (C)\text{supp}_o$.[1]

The behavior of A, from a given configuration $C_o \in C_A$ $(z)C_o \in I_G \times I_Q \quad \forall z \in (C)\text{supp}_o$ is the sequence of configurations

$$C_o, C_o L, C_o L^2, \ldots, C_o L^{\sigma}, \ldots = \{C_o L^{\sigma} | \sigma \geq 0\} \qquad (53)$$

where $L = S \times F$

$$S : V_A \times W_A \to V_A, \quad F : V_A \times W_A \to W_A$$

[1] In fact, L^{σ} is well defined for every $C \in C_A$, but in this study we are only interested in its behavior on initial configurations.

There are several related problems in the construction and applications of DHA's that are formulated in terms of behavior. The generic form of these is as follows.

Given a sequence of functions $\{D_\sigma : Z^k \to \theta, \sigma \geq 0\}$ where θ is finite and contains a distinguished element (called dormant element) denoted by 0 and $(D_\sigma)\text{supp}_o$ is finite for every $\sigma \geq 0$; find a DHA $A = \langle I, \tau \rangle$ with CTM L, $I = Z^k$, an <u>encoding function</u>[1] $h : G \times Q \to \theta$ where $G \times Q$ is the state set of τ, and an initial configuration C_o such that

For

$$(0,0) \in G \times Q$$

$$(0,0)h = 0 \text{ in } \theta \tag{54}$$

$$(z)(C_o L^\sigma)h = (z)D_\sigma \quad \forall\, \sigma \geq 0 \quad \forall\, z \in (D_\sigma)\text{supp}_o \tag{55}$$

In particular, we are interested in the case in which $\theta \_ Q$ (see sections 4.3 and 4.5) i.e., $h : G \times Q \to \theta$ is given by $(\gamma, q)h = q$.

We note that "finding" the DHA $A = \langle I, \tau \rangle$ means determining (if A exists) the structure set Φ, the structure transition function H, the input selector function ψ and the neighborhood function set N. This task is called a <u>synthesis procedure</u> and its outcome is a <u>realization</u>. (see Section 4.3).

---

[1] An encoding function $h:A \to B$ for A, B non-empty sets, is an onto function.

We conclude this section with the introduction of the concept of display.

Let $A = \langle I, \tau \rangle$ be a DHA with $C_A = V_A \times W_A$ (see Section 4.1). Let $\hat{W}_A$ be the set of all sequences $\hat{w}$ of elements of W_A with $|\hat{w}|$ finite. Similarly, let $\hat{V}_A$ be the set of all sequences $\hat{v}$ of elements of V_A with $|\hat{v}|$ finite. For any $v \in V_A$, the partial map

$$\hat{S}_v : \hat{W}_A \to \hat{V}_A$$

defined by

$$w\hat{S}_v = (v,w)S \quad \forall\, w \in W_A$$

for

$$\hat{w} = w_1 w_2, \ldots, w_n \in \hat{W}_A, \tag{56}$$

$$(\hat{w})\hat{S}_v = (v,w_1)S,\ ((v,w_1)S,w_2)S,\ldots,((\ldots((v,w)S,w_2)S,\ldots w_{n-1})S,w_1) \quad \text{(b)}$$

where $L = S \times F$ is the CTM of A, and $\overline{0}$ is the configuration of intensities $(z)\overline{0} = 0 \quad \forall\, z \in I$, is called the <u>display of A with initial structure v</u>.

In (56b) it is understood that the left-hand side is defined if each of the terms in the r.h.s. is defined.

We note that for any $\hat{w} \in \hat{W}_A$, $v \in V_A$

$$|(\hat{w})\hat{S}_v| = |\hat{w}| \tag{57}$$

if $\hat{w}\ \hat{s}_v$ is defined.

The concept of display introduced above, will be used in Chapter 5 in the formulation of the structural identification problem for DHA's.

4.3 Realization Theory of DHA's

In this section we develop and examine an effective procedure for the construction of DHA's. The procedure presented in this section is an adaptation to the particular state structure of the types of DHA's of the usual aglorithm for constructing a minimal automaton given an input-output behavior (see Bobrow and Arbib [1], Eilenberg [2], and Ginsburg [3] for 3 different versions of this procedure).

We start by giving a brief description of the procedure as applied to general finite state automata and then we specialize it to the particular structure of DHA's.

Let A, B be two non-empty finite sets. Assume a function $\lambda: A \times B \to A$ is given. Let B* denote the free monoid with generator B under concatenation of elements of B, i.e., let

$$B^2 = \{bb' \mid \forall\ b, b' \in B\}$$

and

$$B^n = \{\omega b \mid \forall\ \omega \in B^{n-1},\ b \in B\} \text{ for } n > 2\} \qquad (1)$$

then, B* is defined as

$$B^* = B^o \cup B \cup UB^2U, \ldots, UB^n, \ldots, U, \ldots \qquad (2)$$

where $B^o = \Lambda$ is an element not contained in B called the empty word.

We refer to the elements of B* as words. For any $\omega \in B^*$, let $(\omega)\ell = |\omega|$ denote the number of symbols of B concatenated to form ω. In particular $(\Lambda)\ell = 0$. For any ω in B* we refer to $(\omega)\ell$ as the length of ω.

Thus, B* can be characterized as the set of all finite-length words under concatenation of elements of B.

Let $\hat{\lambda}$: A B* → A denote the unique extension of x to B*. $\hat{\lambda}$ is defined inductively as follows:

$$(q,\Lambda)\hat{\lambda} = q \quad \forall\, q \in A$$

$$(q,b)\hat{\lambda} = (q,b)\lambda \quad \forall\, q \in A,\ b \in B \tag{3}$$

$$(q,\omega b)\hat{\lambda} = ((q,\omega)\hat{\lambda},b)\lambda \quad \forall\, q \in A,\ \omega \in B^{*},\ b \in B$$

On A, define a binary relation K as follows:

$$\forall\, q,q' \in A,\ qK'q \iff (q,\omega)\hat{\lambda} = (q',\omega)\hat{\lambda}\ \forall\, \omega \in B^{*} \tag{4}$$

Then we have the following result.

Proposition 1.

(i) K as defined by (4) is a right congruence on A with respect to $\hat{\lambda}$; that is,

$$\forall\, q,q' \in A\ q'Kq \Rightarrow (q,\omega)\hat{\lambda}\ K(q',\omega)\hat{\lambda} \quad \forall\, \omega \in B^{*} \tag{5}$$

(ii) The classes of A under K are effectively computable; that is, there exists a finite sequence of approximating equivalences on A, K_1, K_2, ...,K_M with $K < K_i$[1] $i=1,2,\ldots,$ $K_j \subset K_i$[2] such that
$j>i$

---

[1] $K < K_i$ means that for $q,q' \in A$ $qKq' \Rightarrow qK_iq'$ clearly $<$ is a partial order on the lattice of right congruences on A defined by $\hat{\lambda}$.

[2] We can look at K_j, K_i as subsets of $A \times A$ so that the inclusion operation is well defined.

$$K = \bigcap_{i=1}^{M} K_i \quad \text{i.e., } K = K_M \tag{6}$$

Proof:

See Eilenberg [2] or Ginsburg [3]. For a different statement of the (essentially) same result see Arbib [4].

In Proposition 1 (ii), the equivalence relations K_i i=1,...,M on A are defined as follows:

$$\forall\, q_1, q_2 \in A$$

$$q_1 \, K_i \, q_2 \iff (q_1,\omega)\hat{\lambda} = (q_2,\omega)\hat{\lambda} \;\forall\; , \; (\omega)\ell \leq i \tag{7}$$

$$i=1,2,\ldots,M$$

From this definition we can immediately show the following result:

$$\forall\, q_1, q_2 \in A$$

$$q_1, \, K_{i+1} \, q_2 \iff q_1 K_1 q_2 \text{ and } (q_1,b)\lambda \; K_i \, (q_2,b)\lambda \;\forall\; b \in B \tag{8}$$

It is easy to show (see for instance Eilenberg [,]) that if $K_{i+1} = K_i$ for some i, then $K = K_i$. Further, since A is finite, $i \geq 1$ for which $K = K_i$, is bounded by $|A| - 1$, i.e.,

$$\left.\begin{array}{l} i \,\Big|\, K = K_i \text{ satisfies} \\ \\ \quad i < |A| - 1 \end{array}\right\} \tag{9}$$

As we shall see, the core of our computational procedure is constituted by an alogrithm for computing a congruence of the form of K (as

defined in the context of DHA's) by constructing the sequence K_i of approximating equivalence classes. Condition (9) gaurantees that such a computation is completed in a finite number of steps (i.e., at the most $|A| - 1$ steps).

If we interpret λ as defined above as the state transition function of a given state-output automaton with state set A and input alphabet B, then the construction above, produces a new state output automaton called the quotient automaton with state set $A|K$, input set B and state transition function $\lambda_1 : A|K \times B \to A|K$ defined as follows:

$$A|K = \{(q)E | q \in A\} \tag{10}$$

where $(q)E$ denotes the congruence class defined by K, containing $q \in A$,

$$((q)E,b)\lambda_1 = ((q,b)\lambda)E \quad \forall\, b \in B \tag{11}$$

It is easy to show that λ_1 is well defined i.e., for $q, q' \in (q'')E$

$$((q)E,b)\lambda_1 = ((q')E,b)\lambda_1 \quad \forall\, b \in B \tag{12}$$

Clearly, $|A|K| < |A|$ and the <u>behavior</u>[1] of the original automaton is <u>emulated</u> by the behavior of the quotient automaton in the following sense:

---

[1]The behavior of a state-output automaton with state-set A and input set B is the family of functions $\{M_q : B^* \to A$ for each $q \in A\}$ such that $(\omega)M_q = \hat{\lambda}(q,\omega)$.

Given an initial state $q \in A$ and an input string $\omega \in B^*$, let $(\omega)M_q$ be that state of the original automaton, then the state of the quotient automaton is given by

$$(\omega)M_{(q)E} = ((\omega)M_q)E \tag{13}$$

and clearly, for $b \in B$, $\omega \in B^*$

$$\begin{aligned}(b\omega)\ M_{(q)E} &= [(b\omega)M_q]E = ((\omega)M_{(q,b)\lambda})E \\ &= (\omega)M_{(q,b)\lambda E} = (\omega)M_{(qE,b)\lambda_1}\end{aligned} \tag{13a}$$

Thus, for a given string of inputs the original automaton and the quotient automaton reach homomorphically related states.

The quotient automaton is sometimes called the reduced automaton (it is easy to show that it is unique up to a relabeling of the states).

Now we formulate the state minimization problem outlined above in the context of DHA's. Our first task is to describe the available data. We will consider two modes in which that data is given:

(a) Total mode

(b) Partial mode.

In the total mode (a), we assume that the CTM L of a DHA, $A = \langle I,\tau \rangle$, whose reduced version we want to obtain, is given. This of course is an idealization because

$$L : C_A \to C_A$$

requires an infinite non-countable amount of data points to be fully described (recall that I is unbounded). However, we may assume that what we have is the local description of L (i.e., the state dynamics of τ). Without loss of generality, we may assume that $A = \langle I, \tau \rangle$ is trim, complete, and deterministic with initial state set

$$I_G \times I_Q = (\gamma^o, q^o) \tag{14}$$

Let $\delta : (G \times Q)^{2k+1} \to (G \times Q)$ be the function defined by the formula; for every $C \in C_A$,

$$((z)C, (z + \alpha_{o2})C, \ldots, (z + \alpha_{o,2k+1})C)\delta = (z)(CL) \tag{15}$$

We call δ defined by (15) the <u>isotropic local transition</u> of the DHA A. The vectors $\alpha_{o,i} \in Z^k = I$, $i=1,\ldots,2k+1$ were defined in Section 4.1 (see expresion 4.1-18).

In terms of the local dynamics of A, δ is given by; for $C = (v,\omega)$,

$$((z)v, (z)\omega), ((z + \alpha_{o2})v, (z + \alpha_{o2})\omega), \ldots, (z + \alpha_{o,2k+1})v,$$

$$(z + \alpha_{o,2k+1})\omega)\delta = ((z)v, \omega|(z)v\Psi)H, (\omega|(z)v\Psi)f^{(z)v}) \tag{16}$$

where Ψ is the input selector function of A. Notice that for any $\gamma \in G$,

$$\gamma\Psi = g_i \in N \text{ (the neighborhood set of A where } (z)g_i = (z + \alpha_{i,1}, z + \alpha_{i,2}, \ldots, z + \alpha_{i,m_i}), \{\alpha_{i,R}\}_{R=1,\ldots,m_i} \in Z^k$$

and

$$\{\alpha_{i,1},\ldots,\alpha_{i,m_i}\} - \{\alpha_{o,1},\ldots,\alpha_{o,2k+1}\} \tag{17}$$

for every $g_i \in N$. Thus, the isotropic local transition of A takes, for computing the next state of a type allocated at $z \in I$ all the present states of its nearest neighbors, this justifies the qualifier "isotropic" in the denomination of δ. The objectives behind this construction will become apparent later on in this section.

In the partial mode (b), the available data is not the CTML of A but one or more segments of its orbits.

An orbit of a CTML is a sequence of elements of C_A, C_o, $C_1,\ldots,C_t,\ldots$ such that

$$C_t = C_o L^t \quad t=0,1,2,\ldots \tag{18}$$

A segment of an orbit is a finite sequence of consecutive elements of an orbit (i.e.,

$$\{C_\tau, C_{\tau+1},\ldots,C_{\tau+\sigma} | C_{t+1} = C_t L \quad t = \tau, \tau+1,\ldots,\tau+\sigma-1\}).$$

Given an orbit (or a segment of it) of A, we cannot compute in general the isotropic local transition δ. (Since we only know a segment of L the structure and intensity transition functions of A are only partially known to us.[1]) However, we can compute a relation $\hat{\delta}$, whose graph contains

[1]For instance we know the set Φ but we do not know the structure transition H. Other possibilities can also be handled.

the graph of δ. Next we will give a procedure to find $\hat{\delta}$ given an orbit or a segment of it. This procedure gives us $\hat{\delta}$ in a partially recursive-function representation. Once $\hat{\delta}$ is obtained we obtain from it an approximation $\tilde{\delta}$ to the true isotropic local transition δ.

Let us assume that a segment S of an orbit of L is given i.e.,

$$S = \{C_o,\ldots,C_\tau | C_t \in C_A,\ t=0,\ldots,\tau,\ C_t = C_o L^t,$$

$$(z)C_t \in G \times Q \quad \forall\ z \in I\}^1$$

Clearly, from any consecutive pair of elements C_{t-1}, C_t of S, we can construct a relation $\hat{\delta}_{t-1} : (G \times Q)^{2k+1} \to G \times Q$ defined as follows:

$$(\hat{\delta}_{t-1})\text{Graph} = \{((z)C_{t-1},(z + \alpha_{o,2})\ C_{t-1},\ldots,(z + \alpha_{o,2k+1})C_{t-1};$$

$$(z)C_t) | \forall\ z \in I\} \qquad (19)^2$$

Since C_{t-1}, C_t are elements of C_A, and have finite support. $(\hat{\delta}_{t-1})\text{Graph}$ can be effectively constructed (i.e., we only have to look at the points in $(\overline{C_{t-1}})\text{Supp}_o$). We define the relation $\hat{\delta} : (G \times Q)^{2k+1}\ G \times Q$ or rather its graph, by the formula,

---

[1] $\tau > 0$ is referred to as the horizon of the segment S.

[2] $(\hat{\delta}_{t-1})\text{Graph}$ is a subset of $(G \times Q^{2k+1} \times G \times Q$ in which, a subset of tuples of $(G \times Q)^{2k+1}$ are paired with corresponding elements of $G \times Q$ without repetitions. Thus, since $|(\overline{C_{t-1}})\text{Supp}_o|$ is finite $(\delta_{t-1})\text{Graph}$ contains only finitely many nontrivial tuples.

$$(\hat{\delta})\text{Graph} = \bigcup_{t=1}^{T} (\hat{\delta}_{t-1})\text{Graph} \tag{20}$$

Formally, we write $\hat{\delta}$ as

$$((\gamma_1,q_1),\ldots,(\gamma_{2k+1}, q_{2k+1}))\hat{\delta} = (\gamma_{2k+2}, q_{2k+2})$$

$$(\gamma_i,q_i) \in G \times Q \quad i=1,\ldots,2k+2$$

whenever

$$((\gamma_1,q_1),\ldots,(\gamma_{2k+1}, q_{2k+1}); (\gamma_{2k+2}, q_{2K+2})) \in (\delta_{t-1})\text{Graph}$$

$$\text{for some } t=1,\ldots,T \tag{21}$$

Let δ be the isotropic local transition associated with L, then $\hat{\delta}$ satisfies

$$\hat{\delta} \sqsubseteq \delta \tag{22}$$

that is,

$$((\gamma_1,q_1),\ldots,(\gamma_{2k+1}, q_{2k+1}))\hat{\delta} = ((\gamma_i,q_1),\ldots,(\gamma_{2k+1}, q_{2k+1}))\delta$$

$(\gamma_i,q_i) \in G \times Q \quad i=1,\ldots,2k+1$ if the left hand side is defined.

We define a function $\tilde{\delta} : (G \times Q)^{2k+1} \to (G \times Q)$ as follows:

$$((\gamma_1,q_1),\ldots,(\gamma_{2k+1}, q_{2k+1}))\tilde{\delta} = \begin{cases} ((\gamma_1,q_1),\ldots,(\gamma_{2k+1},q_{2k+1}))\hat{\delta} \text{ if defined} \\ (0,0) \in G \times Q \text{ otherwise} \end{cases} \tag{23}$$

We call $\tilde{\delta}$ the approximation of the isotropic local transition δ determined by the segment S.

If the given data consists of a set of segments $\{S_i, i=1,\ldots,\ell\}$ of the CTM L of A, then it is obvious that the approximation $\tilde{\delta}$ determined by these segments can be obtained from

$$(\tilde{\delta})\text{Graph} = \bigcup_{i=1}^{\ell} \left(\bigcup_{t=1}^{T_i} (\hat{\delta}^i_{t-1})\text{Graph} \right) \tag{24}$$

where T_i is the horizon of S_i and δ^i_{t-1}; $t=1,\ldots,T_i$ are defined by (19). The approximation is then given by (23).

Now we proceed to formulate the state minimization problem for the isotropic local transition δ (or its approximation $\tilde{\delta}$) obtained previously on the basis of the ideas of state minimization of automata discussed at the beginning of the section.

Let $\delta : (G \times Q)^{2k+1} \to G \times Q$ be the isotropic local transition of a DHA A (or a segment-approximation to it) associated with δ we define a family of functions $\{\lambda_j, j=1,\ldots,2k+1)\}$ defined as follows:

$$\lambda_j : (G \times Q) \times (G \times Q)^{2k} \to G \times Q$$

For $(\gamma_1,q_1),\ldots,(\gamma_{2k+1}, q_{2k+1}) \in (G \times Q)^{2k+1}$

$$((\gamma_j,q_j); (\gamma_1,q_1),\ldots,(\gamma_{j-1},q_{j-1}), (\gamma_{j+1},q_{j+1}),\ldots,(\gamma_{2k+1},q_{2k+1}))\lambda_j$$

$$= ((\gamma_1,q_1),\ldots,(\gamma_{2k+1},q_{2k+1}))\delta \qquad j = 1,\ldots,2k+1 \tag{25}$$

Let $((G \times Q)^{2k})^*$ be the free monoid generated by $(G \times Q)^{2k}$ under concatenation and let $\hat{\lambda}_j : (G \times Q) \times ((G \times Q)^{2k})^* \to G \times Q$, be the unique extension of λ_j, $j=1,\ldots,2k+1$.

On $G \times Q$, define the binary relations K^j, $j=1,\ldots,2k+1$ by the following:

$$(\gamma,q),\ (\gamma^1,q^1) \in G \times Q.$$

$$(\gamma,q)K^j(\gamma^1,q^1) \iff ((\gamma,q),\ y)\hat{\lambda}_j = ((\gamma^1,q^1),\ y)\hat{\lambda}_j$$

$$\forall\ y \in ((G \times Q)^{2k})^* \qquad (26)$$

$$j = 1,\ldots,2k+1$$

From proposition 1, we have immediately that K^j is a right congruence on $G \times Q$ with respect to $\hat{\lambda}_j$ for every $j=1,\ldots,2k+1$.

The binary relation on $G \times Q$ defined by

$$K = \bigcup_{j=1}^{2k+1} K^j \qquad (27)$$

is the least upper bound of the congruences K^j, $j=1,\ldots,2k+1$, and consequently a right congruence on $G \times Q$ (see Appendix A.1). We summarize these facts in the following:

Proposition 2.

K, as defined by (27) is a congruence on the algebra $\langle (G \times Q), \{\delta\} \rangle$ of type (2k+1) (see Appendix A.1).

It is clear that for each congruence K^j, j=1,...,2k+1 we obtain a quotient "machine" with state set $(G \times Q)\,|\,K_j$ and state transition function λ_j^1 defined by

$$((\gamma,q)E^j,\ y)\lambda_j^1 = (((\gamma,q),\ y)\lambda_j)E^j \qquad (28)$$

$$\forall\ y \in (G \times Q)^{2k}$$

where $(\gamma,q)E^j$ is the congruence-class on G x Q containing (γ,q), with respect to K_j. We now define the reduced isotropic local transition δ_1 as

$$\delta_1 : (G \times Q/K)^{2k+1} \to G \times_Q K$$

$$((\gamma_1,q_1)E,\ldots,(\gamma_{2k+1},q_{2k+1})E)\delta_1 = (((\gamma_1,q_1),\ldots,(\gamma_{2k+1},q_{2k+1}))\delta)E \qquad (29)$$

Two points are clear; 1) by the first isomorphism theorem[1] there is an epimorphism from the state set G x Q to the state set G x Q/K; 2) since the elements in a given class $(\gamma,q)E$ of (G x Q) K have indistinguishable behavior from the operational point of view they can be merged into a single state. As we shall see shortly, we prefer, for reasons that depend mostly on the applications of DHA's studied in this thesis, instead of merging states, the equivalent operation (from the behavior point of view) of selecting a representative of each class for

[1] See Appendix A.1.

constructing "the reduced DHA" that is a DHA with associated local isotropic quotient transition δ_1.

Towards the end of the section we will describe an algorithm for computing $(G \times Q)|K$ given a DHA with state set $G \times Q$. The central procedure in this algorithm is formed by an efficient routine for computing δ_1 and $(G \times Q)|K$.

Given $(G \times Q)|K$, constructed as indicated in the last few paragraphs we now want to determine G_k, Q_k, the label and intensity sets corresponding to the congruence K. Let $(\gamma,q)E$ be the congruence class (under K) containing the pair (γ,q). We will develop next a computationally-based criterion for choosing one representative $(\gamma^*,q^*) \in (\gamma,q)E$ from each class in $(G \times Q)$ K. Briefly, the criterion is based in an ordering of the elements of each class according to a complexity-index, and choosing from each class $(\gamma,q)E \in (G \times Q)|K$ the representative (γ^*,q^*) with the smallest complexity index.

Let Q be the intensity set of A (the given DHA), G its label set and Φ its structure set. For every $q \in Q$, $q \neq o$, let $(q)n$ be a positive integer called the weight of the intensity q. For every $\gamma \in G$, $\gamma \neq 0$ the weight of γ, $(\gamma)m$ is the positive integer computed by the formula:

$$(\gamma)m = \prod_{\omega \in \mathcal{D}_o f^\gamma} ((\omega) f^\gamma)n \qquad (30)$$

The idea behind the definition in (30) is to somehow capture in $(\gamma)m$ a measure of the time complexity in computing the structure f^γ for some

element $\omega \in \mathcal{D}_o f^\gamma$. We believe that $(\gamma)m$ as defined by (30) gives such a measure and a supportive argument at the intuitive level is provided in the next paragraph.

It is clear that $(\gamma)m$ increases monotonically with the cardinality of $\mathcal{D}_o f^\gamma$. This fact agrees with our intuitive understanding of time complexity of a given function f^γ over a finite set, as a measure of the time required for computing $(\omega)f^\gamma$ for some $\omega \in \mathcal{D}_o f^\gamma$ (perhaps the worst-case ω or the "average"-case ω), because in order to compute f^γ for any $\omega \in \mathcal{D}_o f^\gamma$ we have stored a table of (f^γ)Graph,

$$(f^\gamma)\text{Graph} = (\mathcal{D}_o f^\gamma,\ (\mathcal{D}_o f^\gamma) f^\gamma) \tag{31}$$

and the size of this table is precisely the cardinality of $\mathcal{D}_o f^\gamma$.

In computing ωf^γ, for some $\omega \in \mathcal{D}_o f^\gamma$, we have to do a search[1] in the table of its graph and, although we can do substantially better than linear lexicographic search (see Section 5.1), still the computation time[2] for any of the search procedures available in the literature, (see Aho and Ullman [5] for a survey) and in particular for the one we use in Section 5.1, is determined primarily by the cardinality of $\mathcal{D}_o f^\gamma$. So, $(\gamma)m$ seems to be an adequate criterion for choosing representatives from each of the congruence classes in $(G \times Q)|K$. Specifically, we define

[1]It is easy to see that except for a few special cases, this search operation is the dominant (time-wise) item in the computation of $(\omega)f^\gamma$.

[2]This time is called in the literature, retrieval-time (see Aho and Ullman [5]).

the weight m_K associated with the pair $(\gamma,q) \in (\gamma',q')E$ by

$$(\gamma,q)m_K = \begin{cases} (q)n + (\gamma)m & \text{if } q \neq 0,\ \gamma \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

We assign the weight 0 to the dormant state in order to ensure that this pair is the selected one in the corresponding congruence class.

For each congruence class $(\gamma,q)E$ let (γ^*,q^*) be the pair of label an intensity in $(\gamma,q)E$ such that

$$(\gamma^*,q^*)m_K \leq (\gamma',q')m_K \quad \forall\ (\gamma',q') \in (\gamma,q)E \tag{33}$$

Let G_K, Q_K be the collection of all the representatives of elements in $(G \times Q)|K$ (one for each class). This is the state of our reduced DHA. Let Φ_K be the corresponding set of structures. We denote by $(\gamma,q)E\ rep_G$ the label representative of $(\gamma,q)E$ and by $(\gamma,q)E\ rep_Q$ the intensity representative of $(\gamma,q)E$.

The structure and intensity transition functions H_K and $f_K^\gamma \in \Phi_K$ are given by; for $\gamma^* = (\gamma,q)E\ rep_G$,

$$\omega \in \hat{Q}_K^{2k+1} = \overbrace{(G \times Q|K)^{2k+1}}^{}\, rep_Q,\ |\omega| = n_{\gamma^*} \tag{34}$$

$$\begin{aligned} (\gamma^*,\omega)H_K &= ((\gamma^*,\omega)H,\ (\omega)f^{\gamma^*})E\ rep_G \\ (\omega)f_K^{\gamma^*} &= ((\gamma^*,\omega)H,\ (\omega)f^{\gamma^*})E\ rep_Q \end{aligned} \tag{35}$$

The set of neighborhood functions N_K is given by

$$N_K = G_K \Psi \tag{36}$$

and the input selector function Ψ_K is given by

$$\Psi_K = \Psi|G_K \tag{37}$$

H_K and $f_K^{\gamma^*}$, $\gamma^* \in G_K$ are well defined (by (35)) since the right hand side in (35) corresponds to a transition of the isotropic reduced transition $\delta_1 : ((G \times Q)|K)^{2k+1} \rightarrow (G \times Q)|K$ which is well defined by the first isomorphism theorem (see Appendix A.1).

We note that

$$|G_K \times Q_K| \leq |G \times Q| \tag{38}$$

and that there are morphisms $(G \times Q)|_K \rightarrow G_K$ and $(G \times Q)|_K \rightarrow Q_K$ defined by E rep_G and E rep_Q respectively so that the DHA $A_K = \langle I, \tau_K \rangle$ emulates the DHA $A = \langle I, \tau \rangle$ (τ_K is defined by the expressions (34-37). Further, A_K is complete and trim because δ_1 is a complete function and A is trim.

Theorem 1. The reduced DHA $A_K = \langle I, \tau_K \rangle$ of a given DHA $A = \langle I, \tau \rangle$ is effectively constructible (i.e., algorithmically constructable).

Proof: See algorithm below.

The algorithm that we describe next follows the steps we followed in our discussion towards the description of the reduced DHA of a given

DHA. A block diagram of this algorithm is shown in Figure 1.

Before we analyze the construction algorithm of Figure 1 we make one comment with respect to the complexity measures introduced earlier.

In our application, we consider

$$(q)n = q \quad \forall\, q \in Q \tag{39}$$

The reason for that is that in the realization of DHA that represent epigenetic systems, (see Section 3.5; program IDHA). Lower intensities tend to correspond to simpler structures (i.e., structures with fewer arguments). This implies that for a different complexity criterion, one may obtain a DHA A_K which is isomorphic to the one we obtain from the behavioral point of view, but in which the representative sets G_K and Q_K are different from the ones we obtain. The block diagram of Figure 1 is actually a summary in computational form of the construction of the DHA A_K carried out so far in this section. The only item that remains to be described is the procedure partition which we proceed to describe next.

Procedure partition consists of two subprocedures: Procedure Equivalence and Procedure Sequence and a Function, LIST.

Procedure Equivalence compute the congruence classes $\{(\gamma,q)\ E_i^j\}$ defined by

$$(\gamma,q)K_j^i(\gamma',q') \iff ((\gamma,q),\ \omega)\lambda_j = ((\gamma',q'),\ \omega)\lambda_j$$
$$\forall\ \omega \in ((G \times Q^{2k})^*,\ (\omega)\ell \leq i \tag{40}$$

that is, the approximating sequence for the congruence K_j ($j = 1,\ldots,2k+1$) in a sequential fashion. That is, given the blocks of K_j^i, $\{(\gamma,q)E_i^j\}$, Procedure Sequence is called (see Step 4 in Table 1) to compute the blocks of $K_j^{i+1}\{(\gamma,q)E_{i+1}^j\}$. If no new blocks are created, i.e., $K_j^i = K_j^{i+1} = K_j$ or equivalently, $|\{(\gamma,q)E_i^j\}| = |\{(\gamma,q)E_{i+1}^j\}|$; the procedure halts. Otherwise, Procedure Sequence is called again with K_j^{i+1} as input (i.e., its blocks) to compute K_j^{i+2}. This cycle is continued until $K_j^{i'} = K_j^{i'+1}$ for some $i' < |G \times Q| - 1$.

Procedure Sequence, outlined above, computes, given the blocks of K_j^i, the blocks of K_j^{i+1}, it tests pairwise elements of a block of K_j^i to determine whether they are i+1 indistinguishable. If they are not, two sub-blocks are generated and the elements of the corresponding block are assigned each one to one of the newly generated sub-blocks (Steps 6 to 19). We proceed to compute the assignment to the newly generated sub-blocks of all the elements of the original block creating new sub-blocks as soon as in this assignment i+1-distinguishability is detected.

Procedure Partition, as described in the last paragraph and listed in a form of Algol in Table 1, is an adaptation to our problem of a widely used algorithm for computing blocks defined by equivalence classes on sets ($G \times Q$ in our case) (see for instance, Aho and Ullman [5], Aho, Hopcroft and Ullman [18], Stone [19], etc.). It is well known that for $n = G \times Q$ the order of the number of steps required equals $\sim n \log_2 n \times m$ where $m = |(G \times Q)^{2k}|$.

TABLE 1

Procedure Partition

```
     PROCEDURE PARTITION; J

1    SET (γ,q)E_I^J = G x Q;
2    SET I = 0;
3    SET N = 1;
4    BEGIN SEQUENCE (N,(γ,q)E_I^J, N', (γ,q)E_I+1^J
5    IF N.EQ.N'; HALT; ELSE GO TO 6;
6    N = N'; I = I+1; GO TO 4;

     PROCEDURE SEQUENCE; N; (γ,q)E_I^J; N';
                                (γ,q)E_I-1^J;
1    SET I = 0; M = 0;
2    IF I = L = |(γ,q)E_I-1^J|; HALT; ELSE; I = I+1;
     GO TO 3;
3    SET I_1 = (γ,q)E_I^J;
4    IF I_1 = ∅; GO TO 2; ELSE GO TO 5;
5    PICK (γ,q) ∈ I_1 |(top of stack)*|;
6    SET I_1 = I_1 - {(γ,q)}; I_2 = I_1;
7    SET (γ,q)E_I+1^J = {(γ,q)} M = M+1;
8    IF I_2 = ∅; GO TO 4; ELSE; GO TO 9;
9    PICK (γ',q') ∈ I_2;
10   SET I_2 = I_2 - {(γ',q')};
```

Table 1 (contd.)

11 SET $I_3 = (G \times Q)^{2k}$;

12 IF $I_3 = \emptyset$ GO TO 18; ELSE GO TO 13;

13 PICK $y = I_3$;

14 SET $I_3 = I_3 - \{y\}$;

15 SET S = (y) LIST (γ,q)

16 SET S' = (y) LIST γ',q';

17 IF S = S' GO TO 12; ELSE GO TO 8;

18 SET $(\gamma,q)E^J_{I+1} = (\gamma,q)E^J_I \cup \{(\gamma',q')\}$;

19 SET $I_1 = I_1 - \{(\gamma',q')\}$; IF $I_1 = \emptyset$ GO TO 2; ELSE GO TO 7;

FUNCTION LIST (γ,q)

INPUT; $\gamma,q \in (\gamma,q)C^J_I$; $y \in G \times Q$, $\{(\gamma',q)C^J_I\}$

1 SET $\delta_1 = (\gamma,q, y)\lambda_J$;

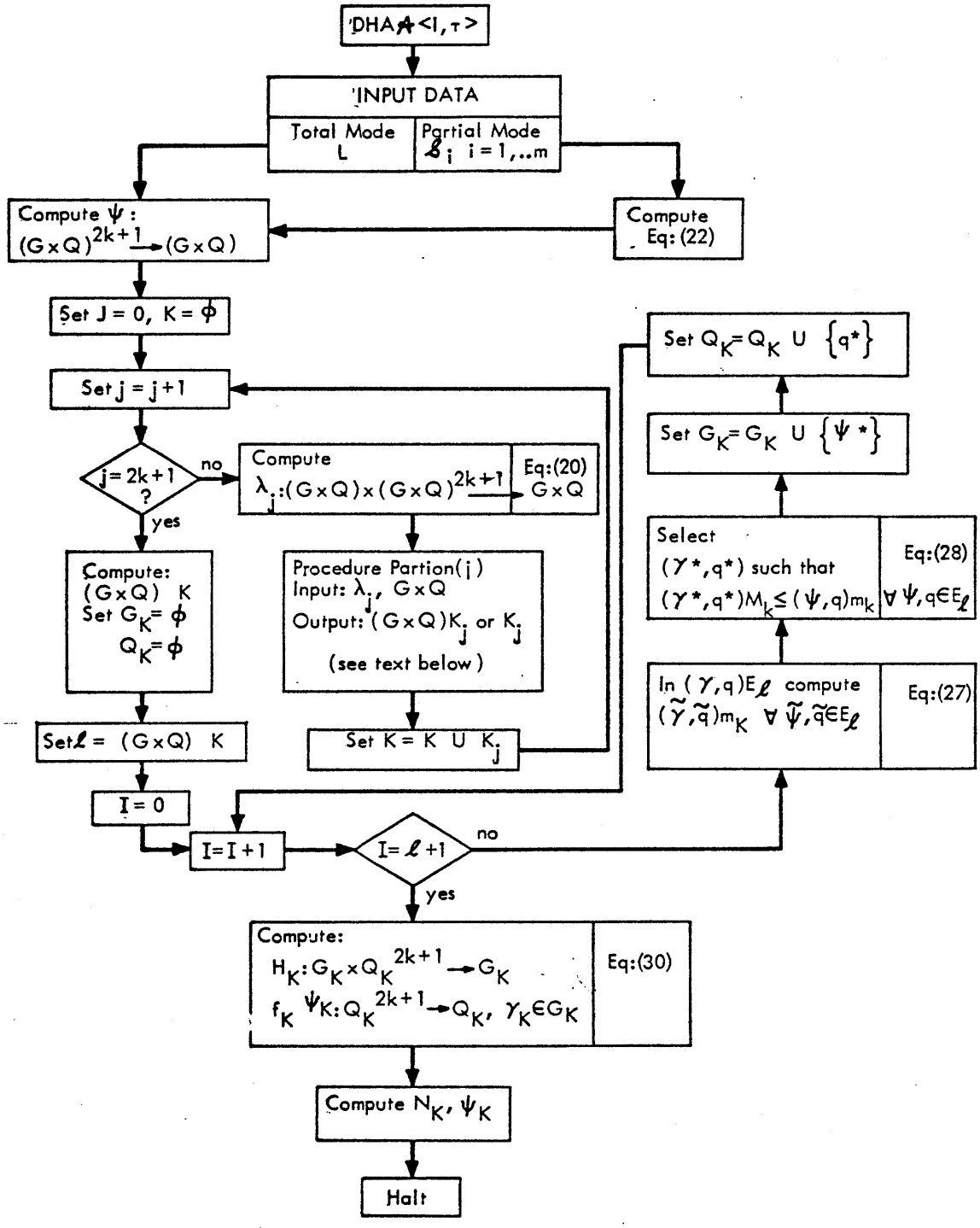
2 SET I = 1;

4 IF $\delta_1 \in (\gamma',q')C^J_I$/* for some (γ',q')* ;

SET i = (y) LIST γ,q; HALT;

ELSE GO TO 5;

5 i = i+1; GO TO 4.



Construction Algorithm

Figure 1

We have introduced in the alogrithm an important variation in order to reduce the number of its steps. This variation consists of the definition of two "floating" sets -- I_1, which contains the elements of the block under test by procedure sequence and I_2, which contains the elements of the block against which each element of I_1 must be tested. As the algorithm proceeds, elements from I_1 which become part of a new sub-block are eliminated from I_1. This reduces the number of tests by an amount proportional to the number of new equivalence classes created in a step K_j^i to K_j^{i+1}. Recall that $|K_j^{i+1}| \geq |K_j^{i+1}|$, thus, in the average we reduce the number of steps per approximation of a given congruence K_j by an amount $K_j/2 = \ell$ so that the number of steps required in the algorithm equals $\sim \frac{n}{\ell} \log_2 n \times m$. In our case, since the congruence classes are rich (many blocks) this is an important reduction, (Step 19 in Table 1 is the updating inside the cycle). In synthesis we reduce the number of tests for new sub-blocks by elminating those elements in the block that remain indistinguishable.

The function (y) LIST γ,q computes the block $(\gamma,q)E_i^j$ to which the pair (γ,q) transfers under an input $y \in (G \times Q)^{2k}$.

The actual implementation of these procedures carried out for the purposes of our study (see Section 5.1) was written in ADEPT, the assembly language of the ADAGE AGT-110 computer; this was done after an attempt to run the algorithm with a Fortran version produced exceedingly inefficient storage-management requirements. Since ADEPT is not a common assembly language we gave, in Table 1, an Algol listing which can easily be

translated to almost any structured higher level language such as PLI or BASIC.

After the blocks of K_j have been computed by Procedure Partition(J), we computed an update of $K = K_{(old)} \cup K_j$ (see Figure 1) so that after 2k+1 we have the blocks of G x Q determined by K. Then we proceed to select from each block, representatives according to the criterion depicted in Equation (33) and finally we construct tables for H_K and $f_K^{\gamma_K} \in \Phi_K$, as indicated in the block diagram of Figure 1.

4.4 Algorithmic Computation of a Covering DHA for a Given Finite Family of DHA's

In this section we develop algorithms for computing several DHA's, each of them a cover for a given family of DHA's

$$M = \{A_1, \ldots, A_s,\ s \geq 0,\ \text{finite}\} \qquad (1)$$

in the sense defined in Section 4.2, i.e., we want to find a DHA A such that $A_i < A$ $i=1,\ldots,s$. The purpose of this exercise is to provide an effective procedure for transforming models of the type developed in Section 3.3[1] into a DHA as described in this chapter.

The advantage in having a DHA that simulates a family of the form of (1) is that we only have to worry about a single type in the study of

[1]We can think of each of the types in Section 3.3 (together with its spatial assignment) as defining a DHA in a 3-dimensional informational space.

the dynamics of the model with a computer simulation and this represents a distinctive advantage with respect to the complexity of the simulation program. Also, the analysis of the dynamic characteristics of the model is facilitated when we only have to consider a single type.

It should be clear from the discussion of Section 4.2 that given $A' = \langle I', \tau' \rangle$ there are an infinite number of DHA's A such that $A' < A$. In this section we discuss two constructions of covering DHA's which are well suited for algorithmic construction:

a) The direct-limit covering DHA

b) The direct-product covering DHA

These two types of covering DHA's share an important characteristic which is desirable for the purposes of our study.

Let A be the direct-limit or direct covering DHA of (1); this implies that there exists a family of morphisms $\{\phi_1^i, \phi_2^i, \phi_3^i, i=1,\ldots,s\}$ where for each i, $\phi_1^i, \phi_2^i, \phi_3^i$ satisfy properties (i)-(vi)[1] of Section 4.2 so that $A_i < A$ $i=1,\ldots,s$, i.e., for each process in A_i, $P_{i,o}^T = \{C_o^i, C_1^i, \ldots, C_t^i, \ldots, C_T^i\}$ there exists a process $P_o^T = \{C_o, C_1, \ldots, C_t, \ldots, C_T\}$ of the same duration in A such that

$$|(C_t^i)\mathrm{Supp}_o| = |(C_t)\mathrm{Supp}_o| \quad t=0,1,\ldots,T \tag{2}$$

[1]That is, ϕ_1^i has the properties of ϕ_1; ϕ_2^i has the properties of ϕ_2; and, ϕ_3^i has the properties of ϕ_3.

$$C_t \text{ proj}_1 \phi_1^i = C_t^i \text{ proj}_1 \quad t=0,1,\ldots,T \tag{3}$$

$$C_t \text{ proj}_2 \phi_2^i = C_t^i \text{ proj}_2 \quad t=0,\ldots,T \tag{4}$$

$$C_t \text{ proj}_1 \Psi \phi_3^i = C_t^i \text{ proj}_1 \Psi^i \quad t=0,\ldots,T \tag{5}$$

where Ψ and Ψ^i are the input selector functions of A and A_i respectively. In short, the morphisms ϕ_1^i, ϕ_2^i, ϕ_3^i are fine morphisms. The importance of this property relies on the fact that the covering operation preserves the time scale and spatial neighboring of the elements two aspects that are fundamental in our study.

With these preliminaries we now proceed to describe the two covering procedures mentioned above

a) Let $M = \{A_1 = \langle I_1, \tau_1 \rangle, \ldots, A_s = \langle I_s, \tau_s \rangle$, $s \geq 1$, finite . Let $\langle Q_i, \Phi_i \rangle$ be the canonical algebra[1] of A_i in M, i=1,...,s. Without loss of generality[2] we may assume that $\langle Q_i, \Phi_i \rangle$ i=1,...,s are complete. The direct-limit covering DHA is constructed as follows:

Let

$$Q^1 = Q_1 \cup Q_2 \tag{6}^3$$

and let Φ^1 be the following set of (structures) operations:

---

[1] See Section 3.4 and Appendix A.1.

[2] See Proposition 4.2-2.

[3] Q_i i=1,2 are the intensity sets of A_i i=1,2.

$$\Phi' = \{f^{\gamma}, \gamma \in G_1 \cup G_2, \tilde{f}^{\gamma} : \mathcal{D}_o \tilde{f}^{\gamma} \to Q^1\} \tag{7}$$

where, by relabeling if necessary, we assume that $G_1 - \{0\}$ and $G_2 - \{0\}$ are disjoint sets. The set $\mathcal{D}_o \tilde{f}^{\gamma}$ is given by

$$\mathcal{D}_o \tilde{f}^{\gamma} = \{\omega \in (\hat{Q}^1)^{2k'+1} | \omega = n_{\gamma}\} \tag{8}$$

with n_{γ} the number of arguments of f^{γ}, $\gamma \in G_1 \cup G_2$ $\gamma \neq 0$, and $k' = \max (k_1, k_2)$ where k_1, k_2 are the dimensions of the informational spaces I_1 and I_2 respectively. The function (operation, structure) $\tilde{f}^{\gamma}$ $\gamma \in (G_1 - \{0\}) \cup (G_2 - \{0\})$ defined by

$$(\omega)\tilde{f}^{\gamma} = \begin{cases} (\omega) f^{\gamma} & \omega \in \mathcal{D}_o f^{\gamma} \\ \emptyset & \text{otherwise} \end{cases} \tag{9}^{1}$$

Notice that for $\gamma \neq 0$, either $f^{\gamma} \in \Phi_1$, or $f^{\gamma} \in \Phi_2$. The dormant structure $\tilde{f}^{o}$ in Φ^1 is defined as

$$\tilde{f}^{o} = \begin{cases} f^{o} \text{ of } \Phi_1 \text{ if } k_1 > k_2 \\ f^{o} \text{ of } \Phi_2 \text{ if } k_2 > k_1 \end{cases} \tag{10}$$

Let $\langle Q^1, \Phi^1 \rangle$ be the algebra defined by (6)-(9). Notice that although

---

[1] We note that $\mathcal{D}_o f^{\gamma} \subset \hat{Q}_1^{2k_1+1}$ if $\gamma \in G_1$, $\mathcal{D}_o f^{\gamma} \subset \hat{Q}_2^{2k_2+1}$ if $\gamma \in G_2$, and $\hat{Q}_j^{2k_j+1} \subset (\hat{Q}^1)^{2k^1+1}$ $j=1,2$.

$\langle Q_1, \Phi_1 \rangle$ and $\langle Q_2, \Phi_2 \rangle$ are complete by assumption, $\langle Q^1, \Phi^1 \rangle$ is not.

Let $H^1 : G^1 \times (\hat{Q}^1)^{2k^1+1} \to G^1$ be the following function, where

$$G^1 = (G_1 - \{0\}) \cup (G_2 - \{0\}) \cup \{0\} \text{ (disjoint union)} \quad (11)$$

for $\omega \in (\hat{Q}^1)^{2k^1+1}$, $\gamma \in G^1$

$$(\gamma,\omega)H^1 = \begin{cases} (\gamma,\omega)H_1 \text{ if } \gamma \in G_1 \ \omega \in \mathcal{D}_o f^\gamma \subset \hat{Q}_1^{2k_1+1} \\ (\gamma,\omega)H_2 \text{ if } \gamma \in G_2 \ \omega \in \mathcal{D}_o f^\gamma \subset \hat{Q}_2^{2k_2+1} \\ 0 \qquad\qquad \text{otherwise} \end{cases} \quad (12)$$

and

$$(0,(0,0,\ldots,0))H^1 = 0 \quad (13)$$

Now we construct a DHA $A^1 = \langle I^1, \tau^1 \rangle$ on the basis of the definitions above and show that A^1 covers A_1 and A_2.

Let

$$I^1 = I_1 \text{ if } k_1 \geq k_2 \text{ or } I^1 = I_2 \text{ if } k_1 \leq k_2 \quad (14)$$

Let $\langle Q^1, \Phi^1 \rangle$ be the canonical algebra of A^1 and let H^1 be its structure transition function we claim that A^1 covers A_1 and A_2 to see that this is the case define

$\phi_1^1 : G^1 \to G_1$ as follows:

$$\gamma\phi_1^1 = \begin{cases} \gamma \text{ if } \gamma \in G_1 - \{0\} \\ \emptyset \text{ if } \gamma \in G_2 - \{0\} \\ 0 \text{ if } \gamma = 0 \end{cases} \quad (15)$$

Similarly, define ϕ_1^2 as follows:

$$\gamma\phi_1^2 = \begin{cases} \gamma \text{ if } \gamma \in G_2 - \{0\} \\ \emptyset \text{ if } \gamma \in G_1 - \{0\} \\ 0 \text{ if } \gamma = 0 \end{cases} \tag{16}$$

Define ϕ_2^i as follows:

$$\phi_2^i : Q^1 \to Q_i \quad i=1,2$$

$$q\phi_2^i = \begin{cases} q \quad q \in Q^i \quad i=1,2 \\ \emptyset \quad \text{otherwise} \end{cases} \tag{17}$$

If $k_1 \geq k_2$, for each neighborhood function g_i in the neighborhood function set N_2 of A_2, define a neighborhood function $\tilde{g}_i$ extended from Z^{k_2} to Z^{k_1} according to 4.2-15 and 4.2-16. The neighborhood function set N^1 of A^1 is then defined as

$$N^1 = N_1 \cup \tilde{N}_2 \tag{18}$$

where $\tilde{N}_2$ is the set of extensions of the neighborhood functions N_2 of A_2 as indicated above.

Let $\Psi^1 : G^1 \to N^1$ be the input selector function of A^1, then from (18), Ψ^1 is given by

$$\gamma\Psi^1 = \begin{cases} \gamma\Psi_1 \quad \text{if } \gamma \in G_2 \\ \tilde{(\gamma\Psi_2)} \text{ if } \gamma \in G_1 - \{0\} \end{cases} \tag{19}[1]$$

[1] ϕ_i i=1,2 are the input selector functions of A_1 and A_2.

If $k_1 < k_2$, we construct the set N^1 as $N_2 \cup \tilde{N}_1$ in the obvious fashion and similarly Ψ^1 is given in this case by

$$\gamma\Psi^1 = \begin{cases} \gamma\Psi_2 & \text{if } \gamma \in G_2 \\ (\tilde{\gamma\Psi_1}) & \text{if } \gamma \in G_1 - \{0\} \end{cases}$$

Let $\phi_3^i : N^1 \to N_i$ i=1,2 be defined by (assuming $k_1 \geq k_2$; the case $k_2 < k_1$ is analogous)

$$g_j^1\phi_3^1 = \begin{cases} g_j^1 \text{ if } g_j^1 \in N_i \\ \emptyset \quad \text{otherwise} \end{cases} \tag{20}$$

$$g_j^1\phi_3^2 = \begin{cases} g_j \text{ if } \tilde{g}_j \in \tilde{N}_2 \\ \emptyset \quad \text{otherwise} \end{cases}$$

It is clear that ϕ_1^i, ϕ_2^i, ϕ_3^i i=1,2 satisfy (i-iv) of Section 4.2[1] and therefore

$$A_i < A^1 \qquad i=1,2 \tag{21}$$

We call A^1, as constructed above the <u>direct limit</u> of A_1 and A_2. We can now replace A_1 and A_2 in M with A^1, since A^1 covers A_1 and A_2. So,

[1]For arbitrary initial sets $I_{G_1} \times I_{Q_1}$ and $I_{G_2} \times I_{Q_2}$ of 1 and 2.

let

$$M_1 = (M - \{A_1, A_2\}) \cup \{A^1\} \tag{22}$$

We now repeat the process for the set M_1; that is, we have $M_1 = \{A^1, A_3, \ldots, A_s\}$ and construct the direct limit A^2 of A^1 and A_3, replace A^1 and A_3 in M_1 to obtain $M_2 = \{A^2, \ldots, A_s\}$ and proceed in the same manner until we obtain A^{s-1}, the direct limit of A^{s-2} and A_s, which is the desired cover. We summarize this construction in the following.

Proposition 1. Let $M = \{A_1, \ldots, A_s\}$ be a finite family of DHA's. Then, there exists a DHA A and an effective procedure to construct it such that A is the direct limit of $\{A_1, \ldots, A_s\}$.

We note that the order in which we construct A from $M = \{A_1, \ldots, A_s\}$ that is, which DHA's in M we take first and which DHA in M we take at each step in the iteration is in material in the sense that all the direct limits of M irrespective of the order in which the iteration in their construction proceeds, are isomorphic (see Section 4.2).

b) Let $M = \{A_1 = \langle I_1, \tau_1 \rangle, \ldots, A_s = \langle I_s, \tau_s \rangle\}$ be a finite family of complete DHA's. Let $\langle Q_i, \Phi_i \rangle$ i=1,...,s be their corresponding canonical algebras.

From $\langle Q_i, \Phi_i \rangle$ i=1,...,s we construct a family of universal algebras $\langle \tilde{Q}_i, \Omega \rangle$ i=1,...,s as follows:

$$1 - \quad \text{Let } \tilde{Q}_i = Q_i \cup \{R\} \quad i=1,\ldots,s \tag{23}$$

where $R > 0$ is an integer such that $R \notin Q_i$ i=1,...,s.

2 - The label set G of Ω is given by

$$G = \{0\} \cup \left(\bigcup_{i=1}^{s} (G_i - \{0\}) \right) \tag{24}$$

In (24) we assume, by relabeling if necessary, that the sets $G_i - \{0\}$, $i=1,\ldots,s$ are mutually disjoint.

3 - We define the algebras $\langle \tilde{Q}_i, \Omega \rangle$ $i=1,\ldots,s$ as follows: for each $\gamma \in G$, let n_γ be the number of arguments of $f^\gamma \in \Phi_j$ for some $1 \leq j \leq s$, and let $\tilde{f}_\gamma$ be an n_γ-ary operation in Ω defined on the sets $\tilde{Q}_i$ $i=1,\ldots,s$ by

$$(\omega)\tilde{f}^\gamma = \begin{cases} (\omega)f^\gamma \text{ if } f^\gamma \in \Phi_i,\ \omega \in \mathcal{D}_o f^\gamma \quad \hat{Q}_i^{2k_i+1} \\ R \qquad \text{otherwise} \end{cases} \tag{25}$$[1]

That is, the operation $\tilde{f}^\gamma$ coincides with the operation f^γ on Q_i if $\gamma \in G_i$ and $\omega \in \hat{Q}_i^{2k_i+1}$. If $\gamma \in G_i$, $\omega \in \hat{\tilde{Q}}^{2k_i+1} - \hat{Q}^{2k_i+1}$ or $\gamma \in G_i$, $\omega \in \hat{\tilde{Q}}_j^{2k_i+1}$ $j \neq i$, we define $(\omega)\tilde{f}^\gamma = R$.

Let ℓ be an integer in $\{1,\ldots,s\}$ defined by

$$\ell = i \in \{1,\ldots,s\} \text{ such that } k_i \geq k_j,\ j \in \{1,\ldots,s\} \tag{26}$$

define

$$\tilde{f}^o \in \Omega \text{ as } \tilde{f}^o = f^o \in \Phi_\ell \tag{27}$$

---

[1] k_i $i=1,\ldots,s$ are the dimensions of the informational spaces I_i, $i=1,\ldots,s$.

Clearly, with these definitions, the set $\{<\tilde{Q}_i,\Omega>,\ i=1,\ldots,s\}$ is a family of similar algebras (see Appendix A.1). Consequently, their direct product, denoted by $\prod_{i=1}^{s} \tilde{Q}_i$, is a well defined algebra with operation set Ω.

In order to convert the algebra $<\prod_{i=1}^{s} \tilde{Q}_i,\Omega>$ constructed above into the form of a canonical algebra, we use the encoding procedure, based on the Cantor pairing function, developed in the proof of proposition 4.2-2. Specifically, let $E_s : \prod_{i=1}^{s} \tilde{Q}_i \to IN$ be the s-iterate of the Cantor pairing function E_2 (see 4.2-19) where IN is the set of natural numbers. Let $Q \subseteq IN$ be the range of E_s. Then $<Q,\Omega>$ is an algebra <u>similar</u> to $<\prod_{i=1}^{s} Q_i,\Omega>$ and <u>isomorphic</u> to it. Specifically, for any $\gamma \in G$,

let $\begin{bmatrix} q_j^1 \\ \vdots \\ q_j^s \end{bmatrix}$ $j=1,\ldots,n_\gamma,\ n_\gamma > 0$

be a set of elements for $\prod_{i=1}^{s} \tilde{Q}_i$. Let $\tilde{q}_j$, $j=1,\ldots,n_\gamma$ be given by

$$\tilde{q}_j = \left(\begin{bmatrix} q_j^1 \\ \vdots \\ q_j^s \end{bmatrix} \right) E_s \qquad j=1,\ldots,n_\gamma \tag{28}$$

Since E_s is 1-1 and onto, $\tilde{q}_j$, $j=1,\ldots,n_\gamma$ in (28) are well defined. The operation $\tilde{f}^\gamma \in \Omega$ is defined on Q by

$$(\tilde{q}_1,\ldots,\tilde{q}_{n_\gamma})\tilde{f}^\gamma = (\tilde{q}_1 E_s^{-1},\ldots,\tilde{q}_{n_\gamma} E_s^{-1})\tilde{f}^\gamma\ E_s \tag{29}$$

$\forall\ (\tilde{q}_1,\ldots,\tilde{q}_n) \in Q$

Thus, $E_s^{-1}: Q \to \prod_{i=1}^{s} \tilde{Q}_i$ is an isomorphism.

Let $\langle Q, \Omega \rangle$ as constructed above be the canonical algebra of a DHA $A = \langle I, \tau \rangle$ with $I = Z^{\ell}, \ell$ given by (26), structure transition function H, neighborhood function N and input selector function Ψ, where H, N and Ψ are given by:

$$H : G \times \hat{Q}^{2\ell+1} \to G$$

$$(\gamma, \omega)H = \sum_{i=1}^{s} (\gamma, \omega \hat{E}_s^{-1} \hat{e}_i^s)\tilde{H}_i \qquad (30)$$

where $\hat{E}_s^{-1}$ is the component-wise extension of E_s^{-1} (see Section 4.2) from $\prod_{i=1}^{s} \tilde{Q}_i$ to $(\prod_{i=1}^{s} Q_i)^{2\ell+1}$ $\hat{e}_i^s$ is the extended projection function (see 4.2-21) $\hat{e}_i^s: (\prod_{i=1}^{s} \hat{Q}_i)^{2\ell+1} \to \tilde{Q}_i^{2\ell+1}$, and the function $\tilde{H}_i$ $1 \leq i \leq s$ is defined from the structure transition function H_i of A_i as follows:

$$(\gamma, \omega \hat{E}_s^{-1} \hat{e}_i^s)\tilde{H}_i = \begin{cases} (\gamma, \omega \hat{E}_s^{-1} e_i^s)H_i & \text{if } \gamma \in (G_i - \{0\}) \; |\omega| = n_\gamma, \\ & \omega \hat{E}_s^{-1} e_i^s \in \hat{Q}_i^{2k_i+1} \\ 0 & \text{otherwise} \end{cases} \qquad (31)$$

We can see from (31) and the disjointedness of the sets $\{G_i - \{0\}$ $i=1,\ldots,s\}$ that for a given γ only one (if any) of the terms in the right hand side of (30) will be different from 0 so H is well defined.

The set of neighborhood functions N is given by all the functions $\tilde{g}_j : Z^{\ell} \to 2^{Z^{\ell}}$, given by (26). Where g_j is the extension of some

$g_j \in N_i$, the neighborhood function set of some A_i i=1,...,s, from k_i to ℓ. (See 4.2-15, 4.2-16 and Figure 4.2-1).

Finally the input selector function $\Psi : G \to N$ is given by:

$$\gamma\Psi = \tilde{g}_j \text{ where } \gamma \in G_i \quad \gamma \neq 0 \quad \gamma\Psi_i = g_j \tag{32}$$

We call the DHA $A = \langle I,\tau \rangle$ constructed above the direct product of the DHA's in the set $M = \{A_1,...,A_s\}$.

Proposition 2. Let $M = \{A_1,...,A_s\}$ be a family of DHA's. Then there exists a DHA $A = \langle I,\tau \rangle$ that is the direct product of $A_1,...,A_s$ and such that $A_i < A$ i=1,...,s.

Proof:

We define the family of morphisms $\{\phi_1^i, \phi_2^i, \phi_3^i, i=1,...,s\}$ by:

$$\phi_1^i : G \to G_i$$

$$\gamma\phi_1^i = \begin{cases} \gamma \text{ if } \gamma \in G_i & i=1,...,s \\ \\ \emptyset \quad \text{otherwise} \end{cases} \tag{33}$$

$$\phi_2^i : Q \to Q_i$$

$$(q)\phi_2^i = q\, E_s^{-1} e_i^s \tag{34}$$

$$\phi_3^i : N \to N_i$$

$$\tilde{g}_j \phi_3^i = \begin{cases} g_j \text{ if } g_i \in N_i \\ \emptyset \text{ otherwise} \end{cases} \tag{35}$$

From (24) we conclude that ϕ_1^i i=1,...,s are well defined and satisfy condition (i) of Section 4.2. Similarly, from (25) and (29) we can see that ϕ_2^i i=1,...,s satisfy condition (ii) finally ϕ_3^i is well defined by the construction of N.

We conclude this section with the following important observation. Despite the apparent complexity of the two procedures studied in this section, both of them are readily implementable as computer programs in the construction of simulation models of the type discussed in Section 3.3, we prefer the direct limit procedure because, as explained in Section 3.4, the model is sequentially specified by the user (program IDHA) and therefore, with a processor with parallel capabilities we can carry out, simultaneously, the specification of the DHA's $\{A_i, \ldots, A_s\}$ and the computation of the direct limit of these DHA's: i.e., as soon as a new DHA is specified, the direct limit of the DHA specified so far and the new one can be computed.

The direct product computation could also be specified as a recursive procedure, but the complexity of the subroutine required to keep track of the number of DHA's already specified makes this recursion a

very inefficient program[1]. On the other hand, if the DHA's $A_1,\ldots,A_s$ are given a priori, the direct-product cover DHA is far less complex[2] to compute then the corresponding direct-limit cover DHA.

Notice that from the algebraic point of view, the DHA's obtained by both procedures are isomorphic.

4.5 Process Control in DHA's

In this section we analyze three special kinds of processes in DHA's

a) Initial Activation Processes

b) Composite Processes

c) Construction Processes

The importance of these three classes of processes in our study resides in the fact that they represent corresponding epigenetic control processes in procaryotes. Therefore, we want to understand their basic characteristics in the context of DHA's.

We discuss the three kinds of processes mentioned above in the context of the formulation of three control problems in DHA's, each one associated with one of these kinds.

a) Let $C : I \to G \times Q$ be a given function where G and Q are the structure label and intensity sets of a DHA $A = \langle I,\tau \rangle$. Let C_A be the set

---

[1]This observation is based on actual experimentation with a version of this recursive implementation.

[2]We measure complexity here as execution time.

of configurations of $\mathcal{A}$. Let $k > 0$ be the dimension of I, and let T_H be a positive integer called the horizon of the DHA (see Section 4.2). In I, we assume as given a closed (finite) subset of points S,

$$S = \{z = (z^1,\ldots,z^k) \in I \mid \alpha^i \leq z^i \leq \beta^i,\ \alpha^i,\ \beta^i \in Z,\ i=1,\ldots,k\} \quad (1)^1$$

we further assume that

$$\overline{(C)\mathrm{Supp}_o} \subseteq S \quad (2)$$

Let $\mathrm{ics}_{\mathcal{A}}$ be the initial configuration set of $\mathcal{A}$ without loss of generality (see Section 4.2)[2], we assume that $\mathcal{A}$ is trim (with respect to T_H) and deterministic. Let $\{(\gamma^o,q^o)\}$ be the initial state set of $\mathcal{A}$.

The initial activation process control problem is stated as follows: find $C_o \in \mathrm{ics}_{\mathcal{A}}$ and a non-negative $t < T_H$ such that[3]

(i) $(C_o)\mathrm{Supp}_o \subseteq S$

(ii) $C_o L^t = C$

(iii) $(C_o L^\tau)\mathrm{Supp}_o \subset S \quad \forall\ \tau \leq t.$

A process P_o^t satisfying (i)-(iii) above, is called and initial activation process associated with C.

---

[1]For instance, in our application of DHA's this region is determined by the allocation of the STOP elements (see Section 3.3).

[2]In particular, propositions 4.2-4 and 4.2-2.

[3]To avoid triviality, we assume that $C \notin \mathrm{ics}_{\mathcal{A}}$.

We have the following:

Lemma 1. It is effectively decidable whether an initial activation process control problem has a solution.

Proof:

Since S is finite, $(C)Supp_o \subseteq S$ $(CL^\tau)Supp_o \subseteq S$ ics_A is finite (by definition, see Section 4.2). We can think on the following algorithm for solving the problem. Run the DHA for each initial configuration $C_o \in ics_A|S^1$ until either for some $t < T_H$, $C_oL^t = C$ or $C_oL^{T_H-1} \neq C$. The algorithm halts whenever and if a C_o is found such that $C_oL^t = C$ $t < T_H$, or all the configuration in $ics_A|S$ have been tested. ▲

It is clear, that the algorithm above, even for one dimensional DHA's is terribly inefficient, i.e., it is basically a sequential enumeration of all instances. Nevertheless, its existence guarantees the well posedness of our problem.

We now study a sequential algorithm for solving the initial activation process whose time complexity is lower than that of the algorithm suggested in Lemma 1.

Let $x_o = \{x_z^o \; \forall \; z \in S\}$ (2a) be a (finite) set of different symbols that we refer to herein as a set of indeterminates in intensity of the o-th step. The first step in the proposed algorithm is to find, for each

[1] $ics_A|S = \{C_o \in ics \; | \overline{C_oSupp_o} \subset S\}$

$z \in S$ the set $(z)\alpha$ of structures of A defined by

$$(z)\alpha = \{\gamma \in G \mid (\gamma,\omega)H = (z)C \text{ proj}_1 \text{ for some } \omega \in \hat{\Omega}^{2k+1}\} \qquad (3)$$

where H is the structure transition function of A. Clearly, if for at least one z, $(z)\alpha = \emptyset$, then the initial activation process control problem has no solution.[1] So, let us assume $(z)\alpha \neq \emptyset \; \forall \; z \in S$.

For every $z \in S$ construct the following subset of $(z)\alpha$

$$(z)\bar{\alpha} = \{\gamma \in (z)\alpha \mid (z)C \text{ proj}_2 \in \text{Range } f^{\gamma}\} \qquad (4)$$

That is, we form $(z)\bar{\alpha}$ by deleting from $(z)\alpha$ all these structures which do not have in their range the intensity of the given configuration at z. Again, if $(z)\bar{\alpha} = \emptyset$ the problem has no solution, so let us assume that $(z)\bar{\alpha} \neq \emptyset \; \forall \; z \in S$.

Let

$$(z)\bar{\alpha} = \{(z)v_1,\ldots,(z)v_{m_z} \in G,\ m_z \text{ finite},\ m_z > 0\} \qquad (5)$$

for some $z \in S$. We form the equations

$$(x_z + \alpha_{m,1},\ldots,x_z + \alpha_{m,\ell_m})f^{(z)v_j} = (z)\ C \text{ proj}_2{}^{2}\ j=1,\ldots,m_z$$

$$\forall\ z \in S \qquad (5a)$$

---

[1] Although a composite process control problem may have a solution as will be discussed later.

[2] i.e., one step backwards in time.

where

$$(z)v_j^{\Psi} = g_m \text{ for some } g_m \in N \tag{6}$$

here, N is theneighborhood set of $\dot{A}$ (see Section 4.1) and

$$(z)g_m = (z + \alpha_m, \ldots, z + \alpha_m, \ell_m) \tag{7}$$

where ℓ_m is the characteristic of g_m.

Now, we have for each z, a finite number of algebraic equations and the next step in our algorithm is to form <u>systems</u> of algebraic equations of the form of (6) and to give a method for solving them. Specifically, we form <u>all</u> the sets of combinations of equations by picking at each $z \in S$, one structure from the corresponding set $(z)\bar{\alpha}$.

The number ξ of different combinations of systems equations that we obtain by the method described above, is given by

$$\xi = \prod_{z \in S} m_z \tag{8}$$

as can be shown by an elementary counting argument, where $m_z = |(z)\bar{\alpha}|$ (see (5)). At first sight it seems that ξ is a very large number. Consider the following case which corresponds to a DHA simulating the lac-operon (see Sections 3.3, 3.5 and 5.3) in which $|S| \simeq 50$. If $m_z = 2$ $\forall$ $z \in S$ then $\xi = 2^{50}$, a very undesirable situation from the computational point of view, and despite the fact that the solution of a system of

[1]See Conway [ ] or Cohn [ ] for a discussion of the general theory of such systems over arbitrary universal algebras.

equations of the form of (6) is relatively easy to obtain, this order of magnitude of the number of systems to be solved would imply that the algorithm is useless from a practical standpoint.

Fortunately, the situation is substantially better than that because of the high selectivity of the types from which the universal type is built (see Sections 3.3 and 4.4), $|(z)\alpha^{-}| = 1$ for most of the points z in S. In addition, recall from Section 3.3 that different types are allocated in specific regions of S and therefore, the structures of the universal type constructed as in Section 4.4 inherit this spatial distribution; a fact that allows us to further reduce the cardinality of each set $(z)\alpha^{-}$. Specifically, suppose that S is divided into regions $S_1,\ldots,S_\mu$

$$S = \bigcup_{i=1}^{\mu} S_i \tag{9}$$

and that associated with each region S_i[1], we have a subset G_i of G of the admissible structure labels in that region; then, if $z \in S_i$, we only have to consider those structures $(z)\alpha^{-}$ such that $(z)v \in G_i$.[2] That is, we only have to write equations for each $z \in S_i$ for the structures in the set

$$(z)\beta = (z)\alpha^{-} \cap G_i \quad \forall\ z \in S_i \tag{10}$$

---

[1]For an example of this spatial organization into regions see Figure 3.3-12 and companion discussion.

[2]We note that the sets G_i are not necessarily disjoint, but they do share (relatively few) some labels in general.

and clearly,

$$|(z)\beta| \leq |(z)\alpha^{-}| \qquad \forall\, z \in S$$

Finally, we can reduce the number of equations of the form of (6) by considering only those that are non-trivial (i.e., equations district of $(x^{o}_{z},\ldots,x^{o}_{2k+1})f^{o} = 0$). In fact, most of the interesting equations will be obtained for those points z such that $z \in \overline{(C)\mathrm{Supp}_{o}}$.

Now we discuss an algorithm for computing solutions to systems of equations of the form of (6). The central observations towards this goal are:

- For any $z \in S$, the indeterminate x^{o}_{z} will appear in at the most, 2k+1 equations. This, as a consequence of the nearest-neighboring interaction requirement.

- The number of equations equals the number of unknowns.

The first observation tells us that each of the systems of equations of the form of (6) can be broken into (not disjoint in general) subsystems of equations. The second, represents a precondition for existence of a solution. Before we pass to the algorithm, we need a few concepts. Let $\langle Q,\Phi\rangle$ be the canonical algebra of A. Let x_{o} be the set of indeterminates. By $\langle Q,\Phi\rangle\,[x_{o}]$, we denote the <u>polynomial</u> algebra of <u>homogeneity</u> 1[1] of the set of indeterminates x_{o} on $\langle Q,\Phi\rangle$. Briefly, this algebra is defined as follows.

---

[1]See Cohn [6], Arbib [4], Lausch and Nobauer [9] and Gratzer [16].

1 - Form a set $Q^{\#}$ of indeterminates disjoint from x_o as follows. For each $q \in Q$, a unique symbol, say $\hat{q}$, is placed in $Q^{\#}$; that is, Q and $Q^{\#}$ are in one to one, onto correspondence. Let $\phi : Q^{\#} \to Q$ be the encoding map.

2 - We now construct a set of sequences of symbols of elements $Q^{\#} U\ x$ and Φ (Φ interpreted as a symbol set, not as an operation set). This set is called the <u>word</u> algebra[1] of Φ over $Q^{\#} U\ x$ and its elements are called words. Associated with every word there is a non-negative integer k called the <u>rank</u> of the word. The word algebra of Φ over $Q^{\#} U\ x_o$, denoted by $W(Q^{\#} U\ x_o, \Phi)$ is constructed sequentially with the rank of the words: Words of rank 0 (zero) are the symbols of $Q^{\#}$, the symbols in x_o and, all $f^{\gamma} \in \Phi$ such that $n_{\gamma} = 0$. Words of rank $\ell + 1$, ($\ell \geq 0$) are all the words of rank ℓ and, for every $f^{\gamma} \in \Phi$ with $n_{\gamma} > 0$, and every sequence of words of rank ℓ of the form $\omega_1,\ldots,\omega_{n_{\gamma}}$ the word $\omega_1,\ldots,\omega_{n_{\gamma}} f^{\gamma}$ is a word of rank $\ell + 1$.[2]

Clearly, from the above construction we have that each word has associated with it a <u>unique</u> minimal rank, and consequently, each word can be decomposed into a finite number of subwords.

We define the polynomial algebra $\langle Q,\Phi\rangle [x_o]$ as the image of the morphism $\hat{\phi}$

---

[1] See Cohn [1] and Birkhoff [17].

[2] $\omega_1,\ldots,\omega_{n_{\gamma}}$ as defined above, are said to be <u>subwords</u> of the word $\omega_1,\ldots,\omega_{n_{\gamma}} f^{\gamma}$. We decreed that a subword ω' of a subword ω'' of a word ω in $W(Q^{\#} U x_o; \Phi)$ is a subword of ω.

$$\hat{\phi} : W(Q^{\#} \cup x_o; \Phi) \to \langle Q, \Phi \rangle [x_o] \qquad (11)^1$$

defined as follows:

$$\left.\begin{array}{l} x_z^o \hat{\phi} = x_z^o \quad \forall\, z \in S \\ \\ Q^{\#}\hat{\phi} = Q \quad \text{1-1 and onto} \end{array}\right\} \qquad (12)$$

It is clear that $\hat{\phi}$ is an epimorphism of algebras if we interpret each word in $W(Q^{\#} \cup x_o)$ as the result of the sequence of operations of the f^{γ}'s that appear in it, in the order in which they appear. The corresponding object in $\langle Q, \Phi \rangle [x_o]$ is an expression.[2] That is, if we assign an element of Q to each indeterminate in the sequence, we obtain as the value of the expression a well defined element of Q.

Remark: The construction of the polynomial algebra $\langle Q, \Phi \rangle [x_o]$ sketched above is an adaptation, for our purposes of a much more general construction of polynomial algebras on algebraic varieties (see Lang [ ] or Cohn [ ]) which is carried out in the context of category theory. For our purposes, the construction given above, which parallels the classical

---

[1] In order for $\hat{\phi}$ to be a morphism of algebras we must convert the set $W(Q^{\#} \cup x_o; \Phi)$ into an algebra. This is easily done by reinterpreting the elements of Φ as operations on words; thus for $\omega_1, \ldots, \omega_{n_\gamma}$ words in $W(Q^{\#} \cup x_o; \Phi)$ $\omega_1, \ldots, \omega_{n_\gamma} f^{\gamma}$ operation equals the word $\omega_1, \ldots, \omega_{n_\gamma} f^{\gamma}$.

[2] Or polynomial.

construction of polynomial rings (in several indeterminates) over an arithmetic ring, is sufficient.

Now, we define the concept of equations on the polynomial algebra $\langle Q,\Phi\rangle [x_o]$. Let W_1, W_2 be two expressions in $\langle Q,\Phi\rangle [x_o]$; an algebraic equation (or equation) is stated formally as

$$W_1 = W_2 \tag{13}$$

with the meaning that if $\{x_1,\ldots,x_k\}$ is a subset of indeterminates of x_o appearing in W_1 and/or W_2, and if we assign to each x_i a value q_i from Q and obtain the same value on both sides of (13), the equality is interpreted in the strict sense and the tuple $(q_1,\ldots,q_k)$ of assignments is said to be a solution of (13).[1]

A system of algebraic equations over $\langle Q,\Phi\rangle [x_o]$ is a (possibly infinite) set of equations

$$a_i = b_i \qquad i \in I \tag{14}$$

where a_i, b_i $\forall$ $i \in I$, are expressions in Q, $[x_o]$ and I is an arbitrary index set. Our objective now is to obtain a criterion for the solvability of a system of the form of (14). (We note that each of the systems of equations defined by (6) are of this form.) We first prove:

---

[1] In the literature (e.g., Lausch and Nobauer [9]) the interpretation given to (13) is more general in the sense that we look for a solution in a suitably defined extension $\tilde{Q}$ of Q. Since in our case the solution must belong to Q we do not consider this, more general, interpretation.

Theorem 1. Let x_o be a given set of indeterminates. Let Λ be the subset of $\langle Q,\Phi\rangle [x_o] \times \langle Q,\Phi\rangle [x_o]$[1] defined by

$$\Lambda = \{(a_i,b_i) \mid a_i = b_i \quad \forall\, i \in I \text{ is a system of equations over } \langle Q,\Phi\rangle [x_o]\} \tag{15}$$

Let Θ, viewed as a subset of $\langle Q,\Phi\rangle [x_o] \times \langle Q,\Phi\rangle [x_o]$ be the congruence on $\langle Q,\Phi\rangle [x_o]$ generated by Λ. Then, for any $g, g' \in \langle Q,\Phi\rangle [x_o]$,

$$g \,\Theta\, g' \iff \text{there exists a chain } g = h_o, h_1, \ldots, h_r = g' \tag{16}$$

of expressions in $\langle Q,\Phi\rangle [x_o]$ such that any two expressions h_j, h_{j+1} are either equal, or h_{j+1} is obtained from h_j be replacing a subword equal to a_i (for some $i \in I$) in the representation of h_j in $W(Q^{\#} \cup x_o;\Phi)$ by b_i, or by replacing a subword of h_j equal to b_i in its representation, by the corresponding a_i.

Proof:

Let $g\ \Theta_1\ g'$ iff there is a chain in $\langle Q,\Phi\rangle [x_o]$ of the form of (16). Clearly, Θ_1 is an equivalence relation on $\langle Q,\Phi\rangle [x_o]$, it is also a congruence for if

$$g_1\ \Theta_1\ g'_1, \ldots, g_{n_\gamma}\ \Theta_1\ g'_{n_\gamma} \quad g_1, \ldots, g_{n_\gamma}, g'_1, \ldots, g'_{n_\gamma} \text{ in } \langle Q,\Phi\rangle [x_o],$$

$n_\gamma > 0$, $f^\gamma \in \Phi$.

---

[1] Direct product of algebras (see Appendix A.1).

Then, for a fixed ℓ, $1 \leq \ell \leq n_\gamma$, $g_\ell = h^\ell_o, \ldots, h^\ell_{r_\ell} = g'_\ell$ implies

$$(g_1,\ldots,g_{\ell},\ldots,g_{n_\gamma})f^\gamma = (g_1,\ldots,h_o,\ldots,g_{n_\gamma})f^\gamma$$

$$= (g_1,\ldots,h^\ell_{r_\ell},\ldots,g_{n_\gamma})f^\gamma$$

$$= (g_1,\ldots,g'_\ell,\ldots,g_{n_\gamma})f^\gamma$$

so that,

$$(g_1,\ldots,g_\ell,\ldots,g_{n_\gamma})f^\gamma \; \Theta_1 \; (g_1,\ldots,g'_\ell,\ldots,g_{n_\gamma})f^\gamma$$

but by transitivity of Θ_1 this implies

$$(g_1,\ldots,g_\ell,\ldots,g_{n_\gamma})f^\gamma \; \Theta_1 \; (g'_1,\ldots,g'_\ell,\ldots,g'_{n_\gamma})f^\gamma$$

hence, Θ_1 is a congruence. Certainly (by definition, in fact) Θ_1 contains Λ. Thus,

$$\Theta \subseteq \Theta_1 \tag{17}$$

On the other hand, every congruence on $\langle Q,\Phi\rangle [x_o]$ containing Λ must contain Θ_1, by definition of Θ_1, and the second isomorphism theorem (see Appendix A.1). Thus, in particular,

$$\Theta_1 \subseteq \Theta \tag{18}$$

Combining (17) and (18) we obtain $\Theta = \Theta_1$ as desired. ▲

Remark: We note that Theorem 1, is just a formalization of a widely used technique for solving algebraic systems of equations in which we use some of the equations to obtain equivalent forms of expressions in such a

manner as to reduce the number of unknowns in the remaining equations. A particularly well known procedure of this kind is the Gauss-Seidel iterative scheme for solving linear algebraic equations. Theorem 1, as presented here is an adaptation to universal polynomial algebras of a theorem in mathematical logic proposed by Knuth [20] (the theorem gives conditions for testing equivalence of sentences in a propositional calculus).

Now we use Theorem 1 to establish a condition of solvability. First, we need the following concepts (Knuth [20]).

- A congruence Θ on $\langle Q,\Phi\rangle$ $[x_o]$ is <u>separating</u> if for any $q, q' \in Q$

$$q \,\Theta\, q' \Rightarrow q = q' \tag{19}$$

- Let ϕ be a monomorphism from the algebra $\langle Q,\Phi\rangle$ to the similar algebra $\langle Q',\Phi'\rangle$. Then, there exists an algebra Q'' such that $Q'' \approx Q'$, such that Q is a subalgebra of Q''. Without loss of generality, let us assume that $Q \cap Q' = \emptyset$. Then, we get Q'' by replacing every element $q\phi$ $q \in Q$, by $q \in Q$, but do not change any other elements of Q'. This procedure is called an <u>embedding</u> of Q into Q' and the resulting algebra Q'' is called the <u>embedded extension</u> of Q in Q'.

<u>Theorem 2</u>.[1] The algebraic system

$$(x_1,\ldots,x_k)a_i = (x_1,\ldots,x_k)b_i \quad i \in I \tag{20}$$

[1] It turns out that Theorem 2 is true even if Q is not finite, but the arguments required additional machinery and since we do not need the result in that generality for our purposes, we omit the proof.

on the indeterminates $x_1,\ldots,x_k \in x_o$, over the algebra $\langle Q,\Phi\rangle [x_o], |Q|$ finite is solvable iff the congruence Θ on $\langle Q,\Phi\rangle [x_o]$ generated by the set $\Lambda = \{(a_i,b_i) | a_i = b_i \; i \in I\}$ of $\langle Q,\Phi\rangle [x_o] \times \langle Q,\Phi\rangle [x_o]$ is separating.

Sketch of the proof (see Lausch and Nobauer [ ] for a more general version of this theorem).

Suppose that Θ is separating; we can then embed $\langle Q,\Phi\rangle$ into $\langle Q,\Phi\rangle [x_o]/\Theta$. This operation gives us a new algebra $\langle \bar{Q},\Phi\rangle [x_o]$ isomorphic to $\langle Q,\Phi\rangle [x_o]/\Theta$. Let $(x_j)E$ $j=1,\ldots,k$ be the congruence class under Θ, containing x_j, then, since

$$(x_1,\ldots,x_k)a_i = (x_1,\ldots,x_k)b_i \quad i \in I$$

by hypothesis we have

$$(x_1,\ldots,x_k)a_i \; \Theta \; (x_1,\ldots,x_k)b_i \tag{21}$$

and thus

$$((x_1,\ldots,x_k)a_i)E = ((x_1,\ldots,x_k)b_i)E$$

so that

$$((x_1)E,\ldots,(x_k)E)a_i = ((x_1)E,\ldots,(x_k)E)b_i \; \forall \; i \in I \tag{22}$$

Equation (22) implies that $(x_1)E,\ldots,(x_k)E$ is a solution of the algebraic system in $\langle \bar{Q},\Phi\rangle [\bar{x}_o]$. Conversely, suppose that the system is solvable and let $(q_1,\ldots,q_k) \in Q$ be a solution; then, by Theorem 1, for any two expressions g g' in $\langle Q,\Phi\rangle [x_o]$, $g \; \Theta \; g' \Rightarrow (q_1,\ldots,q_k)g = (q_1,\ldots,q_k)g'$ hence, in particular if $g = a$, $g' = b$ are words of 0-rank

in Q, $g \Theta g' \Rightarrow a = b$, i.e., Θ is separating.

We recall that the solution $[x_1]E,\ldots,[x_k]E$ characterized by (22) in Theorem 2, is a solution of the algebraic system $a_i = b_i$ $i \in I$ over the algebra $\langle \bar{Q}, \Phi \rangle [\bar{x}_o]$ and not over the algebra $\langle Q, \Phi \rangle [x_o]$. However, by the embedding construction we have a monomorphism ϕ

$$\phi : \langle Q, \Phi \rangle \rightarrow \langle Q, \Phi \rangle [x_o]/\Theta$$

defined by

$$q\phi = (q)E \quad \forall\, q \in Q \tag{23}$$

and, since Θ is separating, $(q)E$ does not contain any other element (different from q) of $\langle Q, \Phi \rangle$ (although it may contain elements from $\langle Q, \Phi \rangle [x_o]$; that is, expressions of the form $(x_1,\ldots,x_k)a, k > 0$). Thus, a solution to the algebraic system $a_i = b_i$ $i \in I$ over $\langle Q, \Phi \rangle [x_o]$ exists if the morphism ϕ^{-1} is defined for every $(x_j)E$ $j=1,\ldots,k$. Thus, $q_1,\ldots,q_k$, $q_m \in Q$ $m=1,\ldots,k$ is a solution in Q of $a_i = b_i$ if

$$q_m = (x_j)E\ \phi^{-1} \quad \text{for some } j, \quad 1 \leq j \leq k \tag{24}$$

In particular, if ϕ as defined by (23) is also an epimorphism and (24) holds, the solution is unique.

We summarize the observations above in the following.

Corollary. If the monomorphism ϕ, defined by (23) has an inverse for every $(x_j)E \in \langle Q, \Phi \rangle [x_o]/\Theta$ $j=1,\ldots,k$, then, the solution $(x_j)E$, $j=1,\ldots,k$ in $\langle \bar{Q}, \Phi \rangle [\bar{x}_o]$ determines a solution in $\langle Q, \Phi \rangle [x_o]$. If ϕ is also an epi-

morphism, this solution is unique.

▲

Now we apply the general results presented above to the algebraic systems in our algorithm (given by (6)) in order to characterize their solvability.

Recall from Section 4.1, that the CTM L of A is the product of two maps S and F where F the intensity configuration transition map, $F : V_A \times W_A \to W_A$, computes, given the present configuration $C' = (v',\omega') \in C_A = V_A \times W_A$, the next intensity configuration ω of A.

In equation (6) we are given $\omega = (C)proj_2$, (the right-hand side of the equations) and the left-hand side is determined by a set of feasible[1] $v' \in V_A$ each one of them, giving rise to one of these systems of equations.

Let ξ' be the number of different feasible systems of equations of the form of (6), (recall that $\xi' \leq \xi$ where ξ is given by (8)). Let $J = \{j \mid j=1,\ldots,\xi'\}$ be an arbitrary enumeration of these systems, i.e., to each system an identifying ordinal from J is assigned. Let $\langle Q,\Phi\rangle$ be the canonical algebra of A and $\langle Q,\Phi\rangle [x_o]$ the corresponding polymomial algebra with x_o the set of indeterminates defined by (2a) and let

---

[1] That is, those v' such that $v = (C)proj_1 = (v_1,\omega)S$ for some $\omega \in W_A$. Recall that $(z)v = (z)(v';\omega)S = ((z)v',\omega \mid (z)v')H$.

$$\left.\begin{array}{ll} (x_z + \alpha_{m,1},\ldots,x_z + \alpha_{m,\ell_m}) f^{(z)v_j} = (z)\ C\ \mathrm{proj}_2 & \text{(a)} \\ \text{with} & \\ (z)v_j \Psi - g_m & \text{(b)} \end{array}\right\} (25)$$

$$\forall\ z \in S \text{ some } j \in J$$

be the j-th system. Define the subset Λ_j, $j \in J$ of $\langle Q,\Phi\rangle [x_o] \times \langle Q,\Phi\rangle [x_o]$ as

$$\Lambda_j = \{(f^{(z)v_j},\ (z)v) \quad \forall\ z \in S\} \qquad (26)^1$$

If the congruence Θ_j, generated by Λ_j as indicated in Theorem 1, is separating, we obtain a solution. If, on the other hand, $_j$ is not separating we will find that at least for one indeterminate x_z^o, the congruence class $(x_z^o)E$ contains at least two $q, q' \in Q$ $q \neq q'$ which means

$$\begin{array}{l} x_z^o = q \\ \text{and} \\ x_z^o = q' \end{array} \qquad (27)$$

which is clearly incompatible. In either case, the construction of Θ_j suggests a family of algorithms for either finding a solution or finding that the corresponding algebraic system is inconsistent. We spell out explicitly one of such algorithms next.

---

[1] We note that S in (26) plays the role of the index set for the equations in the system.

An algorithm for solving (25) which is well suited for the structure of our algebraic systems is given by the following sequence of steps.

1. Solve all of the equations (if any) of the form of (25) such that $n_{(z)v_j} = 0$ i.e. $f^{(z)v_j}$ is a o-ary operation. This is a very simple task because in this case

$$x_z^o = f^{(z)v_j} = (z)\ C\ proj_2 \tag{28}$$

If for any of these equations, $f^{(z)v_j} \neq (z)\ C\ proj_2$ we halt and the initial activation process control problem associated with the j-th system of equations has no solution.

2. We note that most of the equations corresponding to points $z \in \overline{(C)Supp_o} \backslash S$ are trivial equations of the form

$$(x_z,\ldots,x_{z+\alpha_{o,2k+1}})f^o = 0 \tag{29}$$

so that immediately we have $x_{z+\alpha_{o,i}} \quad i=o \quad i=1,\ldots,2k+1$ by definition of f^o.

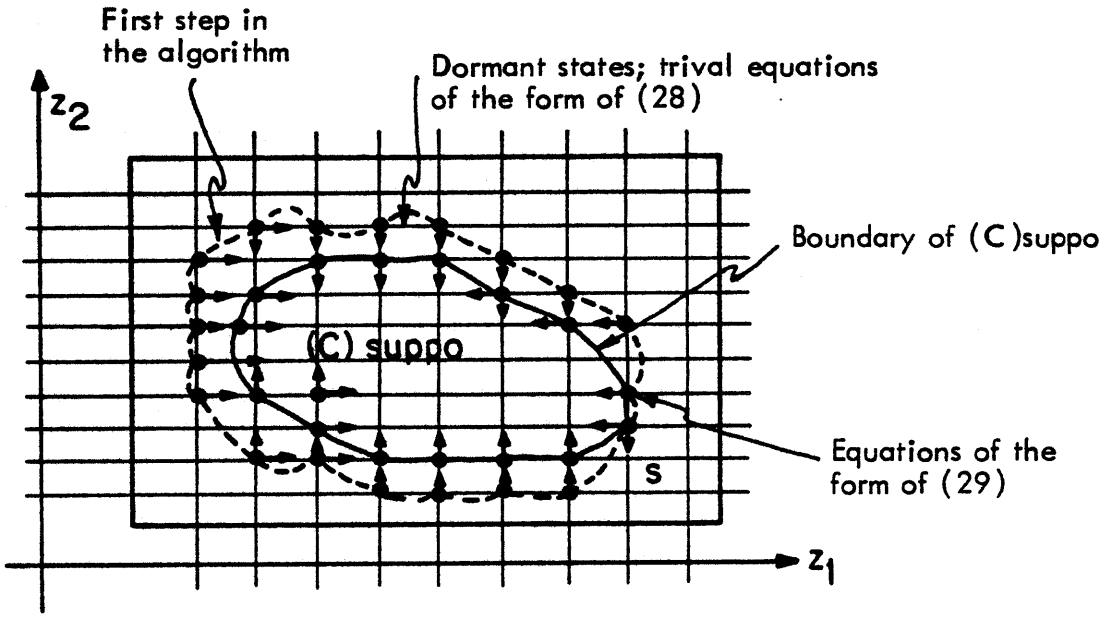
Also if for some z, the corresponding equation in the j-th system is of the form

$$(x_z,\ldots,x_{z+\alpha_{o,2k+1}})f^o = (z)(C)\ proj_2 \neq 0$$

then,

$$x_z = (z)\ C\ proj_2 \tag{30}$$

by definition of f^o.



(→) Arrows indicate the direction of the algorithm for solving an algebraic system of the form of (29)

Illustrative Diagram of Algorithm for Solving Algebraic Systems

of Equations $\{(x_{z+\alpha_{i,1}},\ldots,x_{z+\alpha_i},m_i)f^{(z)v_j} = z(C)\ \mathrm{proj}_2,\ z \in S\}$

Figure 1

Conditions (29) and (30), suggest a start-up procedure for the algorithm for solving each of the systems of equations $j \in J$; namely, we start by solving all the trivial equations at the boundary of $(C)\text{Supp}_o$ and those points belinging to is complement in S. This is schematically illustrated in Figure 1 for a system evolving in a two-dimensional information space.

3. Recall that an indeterminate x_z^o appears in, at the most, 2k+1 equations; thus, the next step in the solution of the system of equations $j \in J$, is to find intensities for those points $z \in (C)\text{Supp}_o$ which have as nearest neighboring points one or more points in the boundary of the set $(C)\text{Supp}_o$ which have many of their unknowns solved in the previous steps. We order our search as follows. We first start with those equations which have a single unknown, then those which have two unknowns, and so on. For instance, suppose that the indeterminate x_z^o appears in two equations as indicated below

$$\left.\begin{aligned} (x_z^o, q_1, q_2) f^{(z)v_j} &= (z)\ C\ \text{proj}_2 \quad (a) \\ (x_{z'}^o, x_z^o, q_3) f^{(z')v_j} &= (z')\ C\ \text{proj}_2 \quad (b) \end{aligned}\right\} \qquad (31)$$

where $q_i \in Q$ i=1,2,3 are previously found solutions for the corresponding unknowns.

In the transition table corresponding to $f^{(z)v_j}$ (31(a)) we search for all tuples (q^1, q^2, q^3) such that

$$(q^1, q^2, q^3)f^{(z)v_j} = (z)\ C\ proj_2$$

and

$$q^2 = q_1,\ q^3 = q_2 \qquad \Bigg\} \quad (32)$$

Suppose that $\tilde{q}^1 = \{q^1_1,\ldots,q^1_m\}$ are the intensities corresponding to q^1 in (32); from these, we select the ones that are compatible with the structure transition function, i.e.,

$$((z)v_j,\ (q^1_{i_k}, q_2, q_3))H = (z)\ C\ proj_1 \quad k=1,\ldots, \qquad (33)$$

say, that these are $(q^1_{i_1}, q^1_{i_2},\ldots,q^1_{i_\ell})$, $q_{i_k} \in \tilde{q}^1$.

Now we test each of these solutions in equation (31(b)) as follows: for each tuple

$$(x^o_{z'}, q^1_{i_k}, q_3)\ k=1,\ldots,\ell$$

we search in the table of $f^{(z')v_j}$ whether there exist tuples $(q, q^1_{i_k}, q_3)$ with $q \in Q$ such that

$$(q, q^1_{i_k}, q_3)f^{(z')v_j} = (z')\ C\ proj_2$$

If there is at least one such tuple then $q^1_{i_k}$ is a feasible solution for x^o_z, otherwise, we discard it. After we have tested all $q^1_{i_k}$ $k=1,\ldots,\ell$, we discard those that do not satisfy (31(b)). The remaining intensities, if any, are feasible solutions for x^o_z. We proceed in this fashion until all the unknowns have been found (or we halt due to an inconsistency). We summarize this procedure in the block diagram of Figure 2. Again, we

remark that because of the high-selectivity of the structures, the number of possible solutions is small (for most points, 1 solution, a few of them , 2 solutions and a very rare case, we found 1 point z with x_z^o having 3 solutions; this is based on simulation of an initial activation control process on the computer implementation of the DHA 5.3).

4. Once we finish step 3 (if successfully) we will obtain a set of configurations $C^1,\ldots,C^m$ such that

$$C^i \in CL^{-1}|S \qquad i=1,\ldots,m \tag{34}$$

Each of these configurations is a candidate for being the configuration C_{t-1} of the initial activation process. If for some C^i, i=1,...,m

$$(z)C^i = (\gamma^o,q^o) \quad \forall\, z \in (C^i)\ \mathrm{Supp}_o$$

we have found a solution for the initial-activation process control problem, else, we repeat the procedure described above with each of the C^i's, to find a C_{t-2} set of feasible configurations. The procedure halts when either

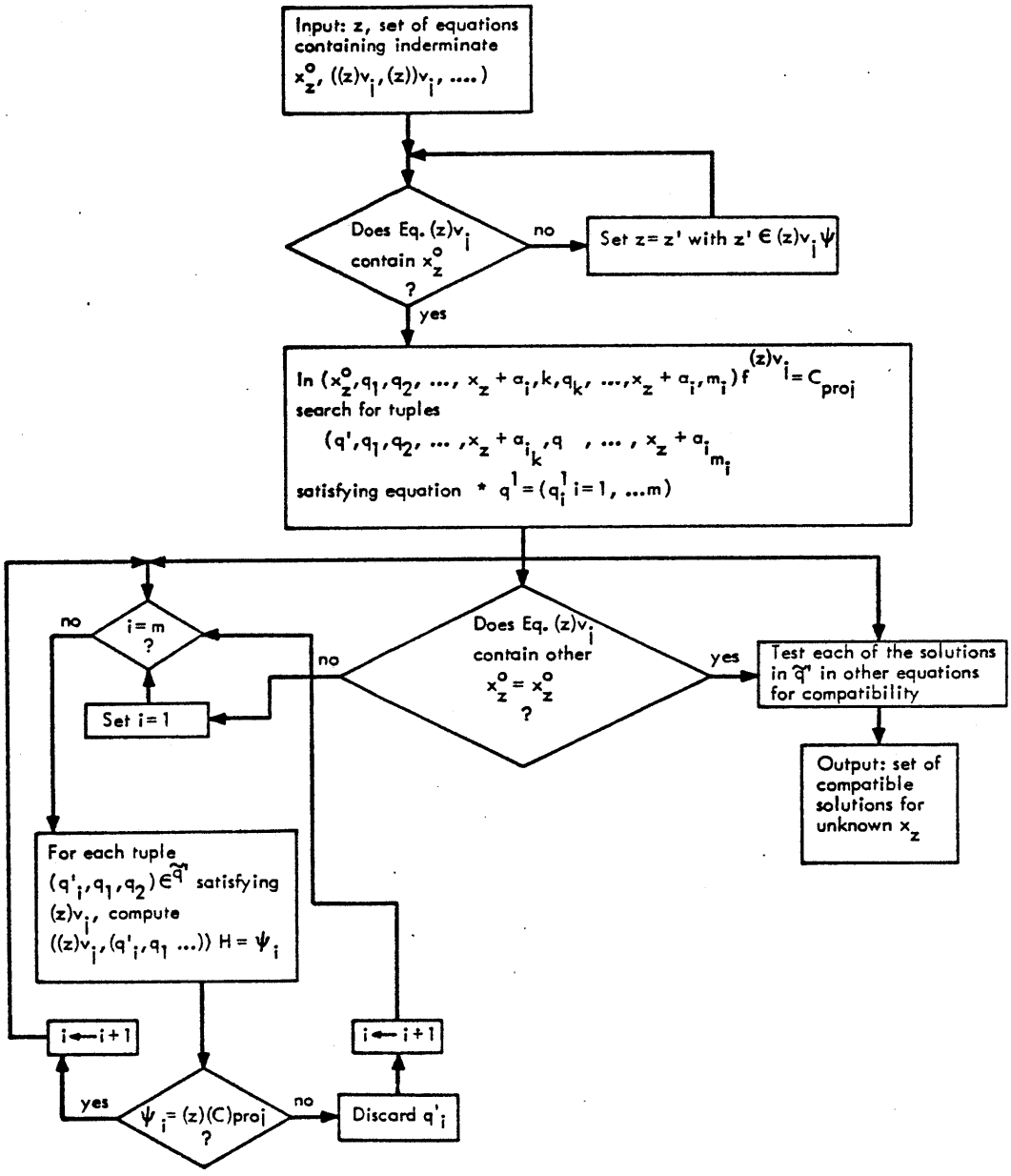
(i) $C^{\ell}_{t-k}\ L^{-1}|S = \emptyset$ for all ℓ some $k \leq T_{H-1}$

(ii) $k = T_{H-1}$

(iii) $(z)C^{\ell}_{t-k} = (\gamma^o,q^o) \quad \forall\, z \in (C^{\ell}_{t-k})\mathrm{Supp}_o,\ C^{\ell}_{t-k} \in \mathrm{ics}_A$

In synthesis, the procedure described above finds an integer $k \leq T_{H-1}$ and a sequence of configurations (if exists)

$$C^i_{t-1},\ C^i_{t-1},\ldots,C^i_{t-k}$$



Block Daigram for Determining Feasible Solutions for V_z^o (see companion text)

Figure 2

such that

$$c^i_{t-k} = CL^{-k}|S$$

$$(z)c^i_{t-k} = (\gamma^o, q^o) \quad \forall\ z \in (c^i_{t-k})Supp_o$$

$$c^i_{t-k} \in ics_A$$

The procedure exploits the local structure of the DHA in the recursive search (in time) for feasible solutions.

In Chapter 5 we will discuss, explicitly, methods for storing the tables of the structure and intensity transition functions so as to make this search efficiently in the context of the structural identification problem (Section 5.2) a problem whose solution is patterned after the procedure discussed here for the initial activation process control problem.

It is important to note that even in the case which $CL^{-1}|S$ is a unique configuration, it may happen that locally, during the execution of the algorithm described in the previous paragraphs, we may obtain several feasible soltuions; however, all of them but the actual solution will be discarded in the manner indicated above, due to the fact that if these local solutions are not global solutions at some point during the procedure we will reach a step in which these local solutions produce inconsistencies.

b) Let $A = \langle I, \tau \rangle$ be a DHA A as in part a). Let $\{c_o, c_1, c_2, \ldots, c_{\tau+1}, \leq T_H\}$ be a sequence of configurations in C_A such that $(c_\sigma)Supp_o \subseteq S$

for $\sigma = 0,1,\ldots,\tau$. Let $U_o, U_1,\ldots,U_\tau$ be a set of functions $U_\sigma : S \to G\times Q$ where G, Q are the label and intensity sets of A. We assume that $U_\sigma \cap C_\tau = \emptyset$ for all $\sigma, \tau, \sigma \neq \tau$. For $\sigma=\tau$, we assume that U_σ and C_τ are disjoint, but (possibly) they interact (see Section 4.1).

A composite process PC_o^τ, with main stream process $\{C_o, C_1,\ldots,C_{\tau-1}, T < T_H\}$ is the sequence of configurations

$$P_o^{T_H} = \{C_o \quad U_o, C_1 \quad U_1,\ldots,C_\tau \quad U_\tau\} \tag{35}$$

with

$$C_{t+1} = (C_t \cup U_t)L \quad t=0,\ldots,\tau \tag{36}$$

where the Union (U) in (35) is disjoint (as defined in Section 4.1).

We remark here that in general,

$$C_{t+1} \neq (C_t)L \cup (U_t)L \tag{37}$$

That is, C_t and U_t, in general, interact (see Section 4.1).

With these preliminaries we are now ready for the formulation of the composite control problem. Let $\{C_o,\ldots,C_{\tau+1}, \tau < T_H\}$ be a given main-stream process in a complete, trim DHA $A = \langle I,\tau\rangle$ with $\overline{(C_T)\text{Supp}_o} \subseteq S$ $t=0,\ldots,\tau$, defined by (1). We want to find a set of functions $U_t : S \to G \times Q$ $t=o,\ldots,$, such that

---

[1]The inequality in (31) becomes an equality only in the case in which C_t and U_t are non-iterative (see Section 4.1).

(i) $C_o \quad U_o \quad ics_A$ (disjoint union)

(ii) $\overline{(C_t \quad U_t)Supp_o} \quad S$

(iii) (36) holds for $t=0,\ldots,\tau$

(iv) For each t, $t=0,\ldots,\tau$ let V_t be the set

$$V_t = \{V_t : S \to G \times Q, \mid \quad (V_t)Supp_o \cap (C_t)Supp_o = \emptyset\} \tag{37}$$

clearly, $U_t \in V_t$ $t=o,\ldots,\tau$. We want to find the U_t's according to the following criterion:

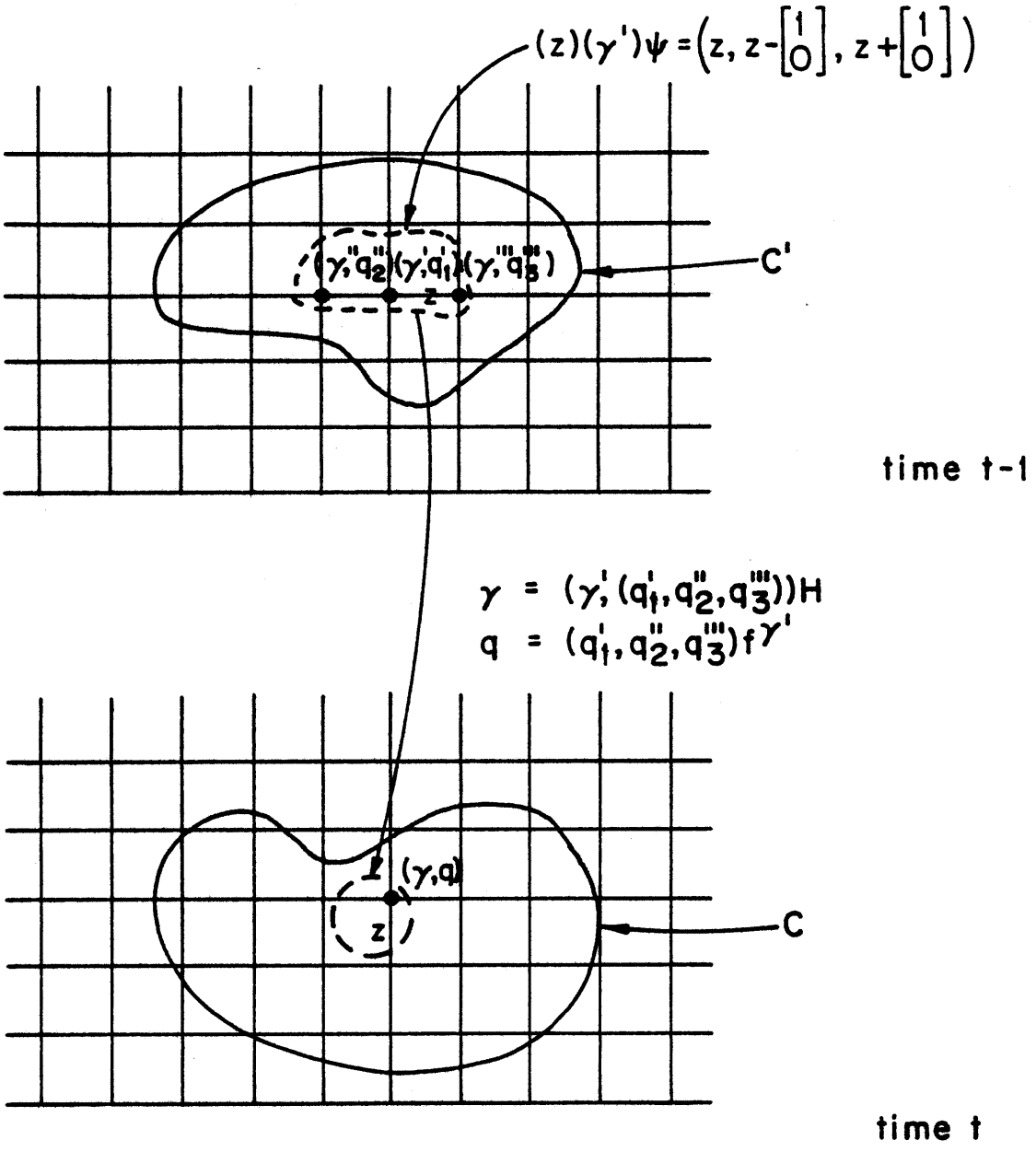
$$\left.\begin{array}{l} \min \left|(U_t)Supp_o\right| \quad t=0,\ldots,\tau \\ \text{s.t.} \\ \qquad U_t \in V_t \end{array}\right\} \tag{39}$$

and (iii) holds.

Condition (iv) is required in order to limit the search for a solution of the problem. We will see, that in our search procedure, this condition comes naturally.

We note that we have <u>not</u> assumed the U_t's to be elements of C_A (i.e., belonging to ics_A or $V_o L^{\sigma} = U_t$ for some $V_o \in ics_A$). In the context of the applications studied in this thesis, we could consider the U_t's as outcomes (for each t) of constructions processes whose characteristics are discussed below in c).

For those readers who are familiar with the theory of partial differential equations, we point out the similarity between the composite con-



Example of Local Compatability in the Mainstream Process of a Composite Process

Figure 3

trol problem stated above and the Dirichlet problem in PDE's (see for instance Treves [21]).

As in the initial process control activation problem, we seek for a solution for the composite problem in a recursive manner, proceeding backwards in time. We describe the corresponding algorithm next.

1. Set $t = \tau + 1$

2. Let $C = C_t$, $C' = C_{t-1}$

Our first task is to check whether the composite problem is <u>well-posed locally</u> for each t.

Let z be a point in $(C')Supp_o$ such that $z \notin \overline{(C')Supp_o} - (C')Supp_o$. Clearly, for every such point we must have for $(C') = (v'\ \omega')$, and $C = (v, \omega)$,

$$\left.\begin{aligned} &(z)v'\Psi = g_j \text{ some } g_j \in N \\ &(z)g_j = (z + \alpha_{j,1}, \ldots, z + \alpha_{j,m_j}) \end{aligned}\right\} \quad (40)$$

$$\left.\begin{aligned} (z)v &= ((z)v',\ (z + \alpha_{j,1})\omega', \ldots, (z + \alpha_{j,m_j})\omega')H \\ (z) &= ((z + \alpha_{j,1})\omega', \ldots, (z + \alpha_{j,m_j})\omega')f^{(z)v'} \end{aligned}\right\} \quad (41)$$

That is, points z of $(C')Supp_o$ whose state behavior cannot be modified by interaction with types allocated externally to $(C')Supp_o$ must evolve to the corresponding given state $((z)v, (z)\omega) = (z)C$ of the type at z. This is illustrated with an example in Figure 3. If for some z in $(C')Supp_o$ (40) and (41) do not hold, then the composite process control problem

has no solution.

3. For each $z \in (C)\text{Supp}_o$, define the set of structures $(z)\alpha$ as follows:

$$(z)\alpha = \{(z)v_j \in G \mid ((z)v_j, \omega)H = (z)(C)\text{proj}_1, \text{ for some } \omega \in \hat{Q}^{2k+1}\} \tag{42}$$

As in (a), if for any $z \in (C)\text{Supp}_o$, $(z)\alpha \neq \emptyset$, the composite control problem has no solution.

Similarly, as in (a), we can reduce the cardinality of $(z)\alpha$ for each $(z) \in S_i$, by intersecting it with G_i, the set of structure labels admissible in region S_i; so, let

$$(z)\bar{\alpha} = (z)\alpha \quad G_i \qquad z \in S_i \tag{43}$$

4. From each set $(z)\bar{\alpha}$ constructed above, eliminate those structures $(z)v_j$ such that $z \notin (C')\text{Supp}_o - (C')\text{Supp}_p$. Since these are fully determined by the local well posedness condition of step 2. Let $(z)\beta$ be the resultant set for each $z \in \overline{(C)\text{Supp}_o}$.

5. Form all possible combinations of algebraic systems $j \in J$, J and index set,

$$\left.\begin{aligned} &(y_{z+\alpha_{i,1}}, \ldots, y_{z+\alpha_i}, m_i) f^{(z)v_j} = (z)(C)\text{proj}_2 \\ &\qquad\qquad (z)v_j \in (z)\beta \\ &(z)v_j \Psi = g_i \\ &(z)g_i = (z+\alpha_{i,1}, \ldots, z+\alpha_i, m_i) \end{aligned}\right\} \tag{44}$$

Where $y_{z+\alpha_{i,k}}$ is either an indeterminate $x_{z+\alpha_{i,k}}$ from a set of indeterminates x_o[1] or an element of Q. Specifically,

$$y_{z+\alpha_{i,k}} = \begin{cases} x_{z+\alpha_{i,k}} & \text{if} \\ & z+\alpha_{i,k} \notin \overline{(C')\text{Supp}_o} - (C')\text{Supp}_o \\ (z+\alpha_{i,k})C' & \text{otherwise} \end{cases} \tag{45}$$

Clearly, each of the algebraic systems $j \in J$ of the form of (44) are algebraic systems of equations on the polynomial algebra $\langle Q,\Phi\rangle [x_o]$, and therefore, the existence of solutions for each of them is determined by the separability of the corresponding congruence Θ_j on $\langle Q,\Phi\rangle[x_o]$, constructed in Theorem 1.

Further, the iterative scheme whose block diagram is given in Figure 2 can be used for solving each of the algebraic systems of the form of (44). Let $V = \{U_j(v_j, \omega_j)\ j \in J'\}$ be the set of solutions obtained in this manner with

$$v_j : (S - (C')\text{Supp}_o) \to G$$

$$\omega_j : (S - (C')\text{Supp}_o) \to Q \qquad j \in J'$$

[1] See 2(a) and companion discussion.

Clearly for each $j \in J'$

$$(C' \cup U_j)L = C \tag{46}$$

by construction.

6. From the set V defined in the previous step we choose as our control function, U_{t-1} at time t-1, a function U_j^* satisfying

$$\min_{\substack{j \in J' \\ U_j \in V}} |(U_j \text{ Supp}_o)| \tag{47}$$

7. Set $t \Leftarrow t-1$ and go to step 2.

This completes the description of the algorithm for the recursive solution of the composite-process control problem.

We conclude the discussion of the composite control problem with an observation about the instances in which it appears in the context of the study of epigenetic control processes in procaryotes. We recall that we assumed the main stream process as given; this main stream process may represent the evolution of a subset of the reaction systems of a group of operons whose dynamic evolution we know, perhaps be experimental measurements of the type discussed in Sections 2.3 and 5.2. However, it may happen (and it usually does) that the scope of experimental measurements does not cover all the reactions systems in the system under study, thus, we can attempt to determine that part of the evolution of the process by representing the main stream process in a suitably chosen

DHA (i.e., of the type of the one developed in Section 3.3) and solve a composite-process as indicated above.

c) We now pass to formulate the construction-control process and propose an algorithm for its solution. Let A be a deterministic, trim DHA as defined in a). Let $P^T_{t_o}$ be a given process in A with initial configuration $C \in C_A$. A constructor process of $P^T_{t_o}$ is a process $\tilde{P}^{t_o}_o$ in A defined as follows:

(i) The initial configuration of $\tilde{P}^{t_o}_o$, $C_o \in ics_A$ satisfies

$$\overline{(C_o)Supp_o} \quad S$$

(ii) $(C_o L^\tau)Supp_o \quad S \quad \forall\, \tau, \quad 0 \leq \tau \leq t_o$

(iii) $C_o L^{t_o} = C'$

with

$$C' = C \cup C'' \quad \text{(disjoint union)} \tag{48}$$

and

C'' does not pass information to $P^T_{t_o}$ i.e.,

$$C \cup C'' L^\sigma = C L^\sigma \cup C'' L^\sigma \qquad \forall\, \sigma, \quad 0 \leq J \leq T-t_o \tag{49}$$

(iv) $$t_o < T_H - 1 \tag{50}$$

where T_H is the horizon of A.

In short, the constructor process function is to produce the initial configuration of the constructed process $P^{T}_{t_o}$ which evolves once C is constructed independently of the constructor $\tilde{P}^{t_o}_{o}$.

We note that the initial configuration C of $P^{T}_{t_o}$ need not be an element of ics_A.

1. The first step in the proposed algorithm is identical to that of the initial activation control problem; that is, given C, the initial configuration of $P^{T}_{t_o}$ we use one step (in time) of the initial activation control algorithm to determine the set of configurations $\Lambda_c = \{C^i, i=1,\ldots,m\}$, such that

$$(C^i)\mathrm{Supp}_o \subseteq S$$

$$C^i L = C \quad i=1,\ldots,m$$

If Λ_c is not empty, we propagate backwards (in time) each C^i until and if we find for some i a configuration $C^i_o \in ics_A$, $(C^i_o)\mathrm{Supp}_o$ or the algorithm reaches step T_{H-1}.

2. Let us consider the case in which Λ is empty and consider the following subsidiary problem. Does there exist a configuration C" such that the following 3 conditions are satisfied?

$$C'': S \rightarrow G \times Q$$

$\Lambda_{C \cup C''}$ is not empty where

$$\Lambda_{C \cup C''} = \{C \in C_A \mid \tilde{C}\, L = C \cup C''\} \tag{51}$$

and

$$C \cup C'' \; L^{\sigma} = C \; L^{\sigma} \cup C' \; L^{\sigma} \tag{52}$$

We construct candidates for C' as follows:

For each $z \in (C)Supp_o - (C)Supp_o$ assign to z a set of structures $(z)\delta$ and intensities $(z)\varepsilon$ such that for each point $z' \in (C)Supp_o$ such that $(z, z')\rho \leq 1$ (i.e., z', z are nearest neighbors) we have

$$((z')Cproj_1, \; Cproj_2 | (z')(Cprog_j)\Psi))H =$$

$$((z')Cproj_1, \; (C \cup C'')proj_2 | (z')(C \cup C''proj_1)\Psi)H$$

and

$$((C \cup C'')proj_2 | (z')(C \cup C''proj_1)\Psi) f^{(z')Cproj_1}$$

$$= (C)proj_2 | ((z')Cproj_1 \Psi) f^{(z')Cproj_1} \tag{53}$$

with

$$(z)C''proj_2 \in (z)\delta \; , \; (z)C''proj_1 \in (z)\varepsilon$$

$$\forall \; z \in \overline{C \; Supp_o} - \overline{C \; Supp_o}$$

We note that (53) is the local version of Equation (52) for σ=1.

Again, our choice of the elements of the sets $(z)\delta$ and $(z)\varepsilon$ is made be restricting to structures and intensities respectively such that if $z \in S_i$ $(z)\delta$ has only structures of G_i and $(z)\varepsilon$ has only intensities of Q_i (G_i, Q_i are the label and intensity sets associated with S_i (see a)).

We eliminate from $(z)\delta$ (resp. $(z)\varepsilon$) all those structures (resp. intensities) for which (53) is not satisfied. Let $\Lambda_1 = \{C_i'', \; i=1,\ldots,\ell\}$ be

the resultant set of feasible configurations i.e., $(C \cup C")L = CL \cup C"_i L$ $i=1,\ldots,\ell$.

3. Now, we repeat the procedure of Step 2 for $\sigma=2$ i.e., we write the local condition of the form of (53) corresponding to

$$(C \cup C")L^2 = CL^2 \cup C"L^2 \tag{54}$$

and eliminate from $(z)\delta$ and $(z)\varepsilon$ all those structures and intensities corresponding to configurations in Λ_1 for which (54) does not hold. We proceed in this manner for $\sigma=3,\ldots,T-t_o$.

4. At the end of step 3 we will have either a subset of configurations $\Lambda_{T-t_o-1}\{C"_i, i=1,\ldots,\ell'\}$ that satisfy (52) at some iteration, wither $(z)\delta$, or $(z)\varepsilon$ are emptied in which case the construction process has no solution.

5. Once Λ_{T-t_o-1} has been determined we solve an initial activation process control problem for <u>each</u> configuration $C \cup C"_i$ $i=1,\ldots,\ell'$. If for one of them we obtain a C^i_o, such that $(C^i_o)\text{Supp}_o \_ S\ C^i_o \in \text{ics}_A$, we have found a solution for the construction problem.

We note that the initial activation process control problem is a subproblem in both the composite and construction process control problems.

We have shown in the last paragraph that there exists recursive algorithms for the solution of the 3 DHA control problems defined at the beginning of this section. These control processes are idealization of corresponding control processes in the epigenetic apparatus of procaryotes.

In general, a process in the apparatus is a combination of these

3 processes thus, its representation in a DHA, is also a combination. For instance, we can have a composite control process in which each of the control functions U_t is the outcome of a construction process. Clearly, a combination of the algorithms described above for composite and construction control problems will provide an algorithm for solving this problem.

It is important to note that the two central procedures in each of the algorithms described above are structure sorting (i.e., as in "find γ such that $(\gamma,\omega)H = \gamma^1$ for some ω") and recursive solution of algebraic equations of the type of (25). With respect to the former we will give an efficient method based on pseudo random storage and retrieval of the structure and intensity transition function tables in Section 4.5.1 in the context of a computer implementation for DHA's. The procedures given here (see Figure 2) for solving algebraic equations are feasible computation time ~ 1800 sec. in AD-110 for (G x Q) of the order of 50 and $|S|$ of the order of 300 (the numbers correspond to the lac operon process discussed in Section 3.5 and 5.3). We have not tried them in larger processes, but it is easy to see that the complexity of the algorithm displayed in Figure 2 grows polynomically with $|G|$ and exponentially with $|S|$. Thus, this algorithm is not efficient for processes corresponding to more than 16 operons which would correspond to a computation time of about 1 hour in the AD-110 computer. We can improve this situation by exploiting the fact, mentioned earlier of parallel solution of the equations; since each indeterminate x_z^o does not appear in all but at the

most 2k+1 we can modify the algorithm of Figure 2 to solve simultaneously several groups of independent equation (i.e., not sharing common indeterminates) however, this would only decrease the time-complexity of the algorithm by an order of magnitude so, the problem remains to find efficient algorithms for large operon systems. We mention that the AD-110 computer in which the implementation was carried on is about 100 times slower than a machine of the IBM-370-145 type so that these limitations are somewhat weakened if the corresponding algorithms are implemented in a computer of this type. Below, we will discuss another method for solving these algebraic systems which although is not feasible in today's commercially available computers, shows some promise when implemented in a new class of computers (CAPP's context addressable parallel processors (see Foster [ ])) under active development at the present.

We conclude this section with some results about the <u>faithful</u> representability of systems of algebraic equations of the form of (6) or (25) over an algebra $\langle Q,\Phi\rangle [x_o]$ of finite order (i.e., $|Q|$ is finite, $|\Phi|$ is finite), as algebraic equations over a suitably chosen <u>Boolean algebra</u> $\mathcal{B}[x]$, (x is a set of indeterminates).

The purpose of the exercise proposed in the last paragraph is twofold. First, we want to give an explicit characterization to the solutions of algebraic systems of the form of (6) which play a central role in the systematic solution of the 3 control problems discussed above; second, it allows us to propose and algorithm for computing these solutions. Although, as we will show later, this algorithm is not practical for im-

plementation in computers available at the present time, the development of special-purpose-processors with arithmetic-logic-units (ALU) capable of processing bit words of several thousands of bits in length is proceeding very fast so that in the very near future (5-10 years) the algorithm to be proposed here will become feasible in a practical sense when implemented in these processors (see Foster [10] for a description of these processors).

We digress for a moment to give some basic notions about Boolean algebras and equations over Boolean algebras. A Boolean algebra $\mathcal{B} = \langle B, \Phi \rangle$ of finite order is defined as follows.

The type τ of Φ (see Appendix A.1) is given by

$$\tau_\beta = \{2,2,1,0,0\}$$

with corresponding label set

$$G_\beta = \{1,2,3,4,5\}$$

and satisfying the following conditions:

for $\alpha, \beta, \gamma \in \mathcal{B}$

Symmetry

$$(\alpha,\beta)f^1 = (\beta,\alpha)f^1 \qquad (\alpha,\beta)f^2 = (\alpha,\beta)f^2$$

Associativity

$$((\alpha,\beta)f^1,\gamma)f^1 = (\alpha,(\beta,\gamma)f^1)f^1 \qquad ((\alpha,\beta)f^2,\gamma)f^2 = (\alpha,(\beta,\gamma)f^2)f^2$$

Absorption laws

$$(\alpha,(\alpha,\beta)f^1)f^2 = \alpha \qquad ((\alpha,\beta)f^2,\alpha)f^1 = \alpha$$

Distributivity

$$(\alpha,(\beta,\gamma)f^2)f^1 = ((\alpha,\beta)f^2, (\alpha,\gamma)f^2)f^1$$

$$(\alpha,(\beta,\gamma)f^1)f^2 = ((\alpha,\beta)f^1, (\alpha,\gamma)f^1)f^2$$

Identity laws

$$(\alpha,f^4)f^1 = f^4 \quad (\alpha,f^5)f^2 = f^5$$

Complementary laws

$$(\alpha,\alpha f^3)f^1 = f^4 \quad (\alpha,\alpha f^3)f^2 = f^5$$

$$f^4 f^3 = f^5 \qquad f^5 f^3 = f^4$$

De' Morgan laws

$$(\alpha,\beta)f^1 f^3 = (\alpha f^3, \beta f^3)f^2$$

$$(\alpha,\beta)f^2 f^3 = (\alpha f^3, \beta f^3)f^1$$

The operation f^1 is usually (see Birkoff and Bartee [22]) denoted by (and called meet) in infix notation, the operation f^2 is denoted by V (and called join) in infix notation, the operation f^3 is denoted by single quotation mark (i.e., $xf^3 = x'$) and called complementation. Finally, the o-th ary operations f^4 and f^5 are denoted by the numbers 0 and 1 which are assumed to be elements of B. In what follows we will use the infix notation for expressions in $\mathcal{B}$. We will assume herein that $B = \{0,1\}$ since this is the only Boolean algebra we consider. Thus, the tables for V, and ' become:

x	y	x y
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x ∨ y
0	0	0
0	1	0
1	0	0
1	1	1

x	x'
0	1
1	0

A Boolean function $f : B^n \to B$ is defined simply as an expression formed by combinations of the 3 operations defined above.

Notation: Let $x = \{x_1,\ldots,x_n\}$ be a set of indeterminates. By x^A, (a sequence of length n of elements of B, i.e., $A = \{(\alpha_1,\ldots,\alpha_n),\ \alpha_i = 0 \text{ or } 1\}$, we denote a monomial of the form $x_1^{\alpha_1} x_2^{\alpha_2},\ldots,x_n^{\alpha_n}$ where

$$x_i^{\alpha_i} \begin{cases} x_i & \text{if } \alpha_i = 1 \\ x_i' & \text{if } \alpha_i = 0 \end{cases}$$

Theorem 3. (Shannon) For any function $f: B^n \to B$,

$$(x)f = \bigvee_{A \in B^n} (A)f \cdot x^A \tag{55}$$

and this decomposition is unique.

Remark: The form of Theorem 3 is called disjunctive form of (x)f (see Miller [11]). We summarize some basic properties of Boolean functions in

Lemma 2.

(i) $\bigvee_{A \in B^n} x^A = 1$

(ii) $x^A \cdot x^C = 0$ for $A \neq B$ $A, C \in B^n$

(iii) $(\bigvee_{A \in B^n} \alpha_A \cdot x^A) \vee (\bigvee_{D \in B^n} \beta_D \cdot x^D) = \bigvee_{A \in B^n} (\alpha_A \vee \beta_A) \cdot x^A$

(iv) $\alpha_A, \beta_A = 0$, or 1 for all $A \in B^n$

(v) $(\bigvee_{A \in B^n} \alpha_A x^A) \quad (\bigvee_{D \in B^n} \beta_D x^D) = \bigvee_{A \in B^n} \alpha_A \beta_A \, {}^A$

(vi)
$$\alpha^\beta = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{of } \alpha \neq \beta \end{cases} \qquad \alpha, \beta \in B$$

(vii)
$$A^D = \begin{cases} 1 & \text{if } A = D \\ 0 & \text{if } A \neq D \end{cases} \qquad A, D \in B^n$$

An equation in one indeterminate over $\mathcal{B}$ is an expression of the form

$$(x)f = (x)h \tag{56}$$

with f, h functions in one variable over B.

Theorem 4. Every equation over $\mathcal{B}$ in one indeterminate x can be written in the form

$$\alpha_1 \cdot x \vee \alpha_2 \cdot x' = 0$$

Proof:

Note first that there are 4 different functions in one indeterminate over $\mathcal{B}$: $(x)g_1 = 0$, $(x)g_2 = 1$, $(x)g_3 = x$ and $(x)g_4 = x'$.

Thus, the form $\beta_1 x \vee \beta_2 x'$ can be made to represent any $(x)g_i$ $i=1,\ldots,4$ by a proper choice of β_1 and β_2 (Eq $\beta_1=1=\beta_2 \Rightarrow \beta_1 x \vee \beta_2 x' = x \vee x' = 1$, $\beta_2 = 0$ $\beta_1 = 1$, $\beta_1 x \vee \beta_2 x' = x$, and so on).

From (56) we have that

$$((x)f)' = ((x)h)' \tag{57}$$

(from now on we write $((x)f)'$ as $(x)f'$ so, "multiplying"[1] (56) by $(x)f'$ on both sides,

$$0 = (x)f(x)f' = (x)h(x)f'.$$

Similarly, multiplying (57) by $(x)f$,

$$(x)h'(x)f = ((x)h')(x)f = 0$$

Thus, adding[2] these two equations we have

$$(x)f(x)h' \vee (x)f'(x)h = 0 \tag{58}$$

---

[1] Meet $\equiv$ multiplication.

[2] Addition $\equiv$ join.

But $(x)f = \alpha_1 x \vee \alpha_2 x'$ for some $\alpha_1, \alpha_2 \in \mathcal{B}$ and

$(x)h = \beta_1 x \vee \beta_2 x'$ for some $\beta_1, \beta_2 \in \mathcal{B}$

so (58) becomes (using De'Morgan laws, see Miller [ ])

$$(\alpha_1\beta_1'\beta_2' \vee \beta_1\alpha_1'\alpha_2' \vee \alpha_1\beta_1' \vee \beta_1\alpha_1')x \vee (\alpha_2\beta_1'\beta_2' \vee \beta_2\alpha_1'\alpha_2' \vee \alpha_2\beta_2' \vee \beta_2\alpha_2')x' = 0$$

which is of the desired form. ▲

Let $\leq$ be a binary relation on $\mathcal{B}$ defined by $\alpha \leq \beta \Rightarrow \alpha \cdot \beta = \alpha$. Clearly, $\leq$ defines a partial ordering on $\mathcal{B}$.

Now we characterize solutions of an equation $(x)f = 0$ over B. (notice that Theorem 4 allows us to consider just this case.)

Lemma 3. The following statements are equivalent.

(i) $\alpha x \vee \beta x' = 0$

(ii) $\beta \leq x \leq \alpha'$

(iii) $x = \alpha' x \vee \beta x'$

Proof:

(i) → (ii) by definition of the join operation, $\alpha x \vee \beta x' = 0$ implies $\alpha x = 0$ and $\beta x' = 0$. But $\alpha x = 0 \Rightarrow (\alpha x)' = 0' = 1$ or $\alpha' \vee x' = 1$ thus $(\alpha' \vee x') \cdot x = 1 \cdot x$ and $\alpha' x = x \Leftrightarrow x \leq \alpha'$.

Similarly, $\beta x' = 0 \Rightarrow (\beta x')' = 1$, or, $\beta' \vee x = 1$. Thus, $(\beta' x \vee x)\beta = \beta$ or $\beta x = \beta \Leftrightarrow \beta \leq x$.

(ii) → (iii) from (ii) we have $\alpha' x = x$, $\beta x' = 0$ thus, $\alpha' x \vee \beta x' = x \vee 0 = x$

(iii) → (i) let $x = \alpha' x \vee \beta x'$* then, multiplying * by x',

$$0 = \beta x'$$

Multiplying * by αx,

$$\alpha x = \alpha\beta x' = 0$$

thus, $\alpha x + \beta x' = 0$ as desired.

▲

Remark: See Abian [12] for a more general form of the Lemma in arbitrary Boolean algebras.

Now, we give a criterion of well-posedness for an equation of $\mathcal{B}$ in one indeterminate.

Theorem 5: (Rudeanu [13])[1] $\alpha x \veebar \beta x' = 0$ over $\mathcal{B}$ has a solution iff $\alpha\beta = 0$.

Proof:

This equation is equivalent to

$$\beta \leq x \leq \alpha'$$

From Lemma 3, thus, $\beta \leq \alpha'$ (transitivity of $\leq$) or

$$\beta\alpha' = \beta \tag{59}$$

[1]The proof given here is a special case of the one proved by Rudeanu [13]. See also Abian [12].

Multiplying both sides of (59) by α we get $\alpha\beta = 0$ as desired. Conversely, let $\alpha\beta = 0$. Then, $\alpha\beta \veebar \beta\beta' = \alpha\beta \vee 0 = 0$ which means that $x = \beta$ is a solution.

Corollary (Rudeanu [ ]) If the equation $\alpha x \veebar \beta x' = 0$ has a solution the all its solutions are given by

$$x = \beta \veebar \alpha' y \qquad \forall\, y \in B$$

The previous theorem and corollary are the basis for the recursive algorithm for solving algebraic equations over $\mathcal{B}$ to be developed below. First, we generalize the results presented for the algebraic equation over $\mathcal{B}$ in one indeterminate to equations over several indeterminates.

Theorem 6.

(i) Let

$$(x)f = (x)g \tag{60}$$

be an algebraic equation over $\mathcal{B}$ with $x = \{x_1,\ldots,x_n\}$ $n>1$. Then there exists a function $(x)F$ on $\mathcal{B}^n$ such that each solution of $(x)F=0$ is a solution of (60) and vice versa.

(ii) Let

$$(x)f_i = (x)g_i \quad i \in I \tag{61}$$

I an index set, be an algebraic system over $\mathcal{B}$, then there exists a function $F : B^n \to \mathcal{B}$ such that each solution of $(x)F = 0$ is a solution of (61) and viceversa.

Proof:

(i) From (60) we have

$$(x)f' = (x)g' \tag{62}$$

Thus, as in Theorem 4, multiplying (60) by $(x)g'$, multiplying (62) by $(x)g$ and adding the resultant equations we have

$$(x)F = (x)f'(x)g + (x)f(x)g' = 0$$

which is of the desired form.

(ii) By (i), each equation in (61) can be written as $(x)F_i = 0$ $i \in I$ thus

$$(x)F = V_{itI}(x)F_i$$

by definition of V.

By Theorem 6, we can restrict ourselves to study the solutions of a single equation of the form

$$(x)F = 0 \tag{63}$$

$F : \mathcal{B}^n \to \mathcal{B}$ $x = x_1,\ldots,x_n$ a set of indeterminates.

Theorem 7. (Boole [23])[1] Let $F : \mathcal{B}^n \to \mathcal{B}$ be a Boolean function. The equation $(x)F = 0$ is well posed (consistent) iff

$$\prod_{A \in \mathcal{B}^n} (A)F = 0 \tag{64}$$

[1]The proof to this result originally given by Boole himself is based on some axioms of propositional calculus; here we offer an algebraic proof patterned after Boole's proof.

Proof:

Assume (60). From Theorem 3 we deduce immediately

$$(A)F \leq \bigvee_{A \in \mathcal{B}^n} (x)F$$

and since $(x)F = 0$, (64) follows.

Conversely, assume (64). The case n=1 was proved in Theorem 5 (see also Rudeanu [13]). Assume that the hypothesis of the theorem has been proven for all $k<n$. Define

$$(x_1,\ldots,x_{n-1})h = (x_1,\ldots,x_{n-1}1)F \quad (x_1,\ldots,s_{n-1},0)F$$

then (64) implies

$$\Pi(\alpha_1,\ldots,\alpha_{n-1})h = \Pi(\alpha_1,\ldots,\alpha_{n-1},1)F \ (\alpha_1,\ldots,\alpha_{n-1},0)F$$

$$\Pi(\alpha_1,\ldots,\alpha_{n-1}) \in \mathcal{B}^{n-1} \quad \Pi(\alpha_1,\ldots,\alpha_{n-1}) \in \mathcal{B}^{n-1}$$

$$= \Pi_{A \in \mathcal{B}^n} (A)F = 0$$

Thus, by the induction hypothesis, the equation $(x_1,\ldots,x_{n-1})$ has a solution. Let this solution be $x_i = \beta_i$ $i=1,\ldots,n-1$. Thus,

$$(\beta_1,\ldots,\beta_{n-1},1)F \cdot (\beta_1,\ldots,\beta_{n-1},0)F = 0$$

But then, by Theorem 5, the equation

$$(\beta_1,\ldots,\beta_{n-1},1)F \ x_n \vee (\beta_1,\ldots,\beta_{n-1},0)x'_n = 0 \qquad (65)$$

has a solution, say $x_n = \beta_n$. But, by Theorem 3, for fixed $(x_1,\ldots,x_{n-1})$

$$(x_1,\ldots,x_{n-1},x_n)F = (x_1,\ldots,x_{n-1})F\ x_n \vee (x_1,\ldots,x_{n-1},0)Fx_n'$$

Hence, (65) implies

$$(\beta_1,\ldots,\beta_{n-1},\beta_n)F = (\beta_1,\ldots,\beta_{n-1},1)F\ \beta_n \vee (\beta_1,\ldots,\beta_{n-1},0)F\beta_n' = 0$$

which means that $(\beta_1,\ldots,\beta_{n-1},\ _n)$ is a solution of $(x)F = 0$.

Now we are ready to formulate our algorithm. Let $(x)F = 0$ be the given equation over $\mathcal{B}$[1] $F : \mathcal{B}^n \to \mathcal{B}$. Write $(x_1,\ldots,x_n)F = 0$ as

$$(x_1,\ldots,x_{n-1},1)F\ x_n \vee (x_1,\ldots,x_{n-1},0)F\ x_n' = 0 \qquad (66)^2$$

Now, by Theorem 5, this equation has a solution iff

$$(x_1,\ldots,x_{n-1})h_{n-1} = (x_1,\ldots,x_{n-1},1)F \quad (x_1,\ldots,x_{n-1},0) \qquad (67)$$

and by Lemma 3 (ii) this solution is bounded as

$$(x_1,\ldots,x_{n-1},0)F \leq x_n \leq (x_1,\ldots,x_{n-1},1)F' \qquad (68)$$

and by Lemma 3 (iii) this solution satisfies.

$$(x_n = (x_1,\ldots,x_{n-1},1)F'\ x_n + (x_1,\ldots,x_{n-1},0)F\ x_n' \qquad (69)$$

---

[1]If instead of $(x)F = 0$ we are given an algebraic system $(x)f_i = (x)g_i$ $i \in I$ we convert it to the form $(x)F$ as indicated in Theorem 6.

[2]This is possible since by Theorem 3, for <u>fixed</u> $x_1,\ldots,x_{n-1}$ $(x_1,\ldots,x_n)F$ has the form of (66).

for fixed $(x_1,\ldots,x_{n-1})$.

From (67) we have

$$(x_1,\ldots,x_{n-1})h_{n-1} = 0$$

This equation can be written as

$$(x_1,\ldots,x_{n-2},1)h_{n-1}x_{n-1} \vee (x_1,\ldots,x_{n-2},0)h_{n-1}x'_{n-1} = 0$$

which has a solution (for x_{n-1}) bounded as

$$(x_1,\ldots,x_{n-2},0)h_{n-1} \leq x_{n-1} \leq (x_1,\ldots,x_{n-2},1)h'_{n-1} \tag{70}$$

iff

$$(x_1,\ldots,x_{n-2} = (x_1,\ldots,x_{n-2},0)h_{n-1}\ (x_1,\ldots,x_{n-2},1)h_{n-1} = 0 \tag{71}$$

We note that

$$(x_1,\ldots,x_{n-2})h_{n-2} = \Pi(x_1,\ldots,x_{n-2},\alpha_{n-1},\alpha_n)F \tag{72}$$
$$(\alpha_{n-1},\alpha_n) \in B^2$$

Clearly, from the discussion above, we can generalize so that at the jth step of the procedure, we obtain an equation of the form

$$(x_1,\ldots,x_{n-j})h_{n-j} = \Pi(x_1,\ldots,x_{n-2},\alpha_{n-j+1},\ldots,\alpha_n)F = 0 \tag{73}$$
$$(\alpha_{n-j+1},\ldots,\alpha_n) \in B^j$$

which can be written as

$$(x_1,\ldots,x_{n-j})h_{n-j} = (x_1,\ldots,x_{n-j-1},1)h_{n-j}\ x_{n-j} \vee (x_1,\ldots,x_{n-j-1},0)$$

$$h'_{n-j}x'_{n-j} = 0$$

which has a solution bounded as

$$(x_1,\ldots,x_{n-j-1},0)h_{n-j} \leq x_{n-j} \leq (x_1,\ldots,x_{n-j-1},1)h'_{n-j} \qquad (74)$$

iff

$$0 = (x_1,\ldots,x_{n-j-1},0)h_{n-j}(x_1,\ldots,x_{n-j-1},1)h_{n-j}$$

$$= (x_1,\ldots,x_{n-j-1})h_{n-j-1} \qquad (75)$$

Thus, in particular at the n-th step of the procedure, we have an equation of the form

$$(x_1)h_1 = 0$$

with solution bounded as

$$(0)h_1 \leq x_1 \leq (1)h'_1$$

iff

$$(0)h_1(1)h'_1 = 0$$

with

$$(x_1)h_1 = \prod_{(\alpha_2,\ldots,\alpha_n)\ \in\ B^{n-1}} (x_1,\alpha_2,\ldots,\alpha_n)F$$

recall from the corollary in Theorem 5 that if $(x_1)h_1 = 0$ has a solution, all its solutions are of the form

$$x_1 = (0)h_1 \vee (1)h_1' y \quad \forall y \in B \tag{76}$$

Thus, we have found an explicit expression for all the solutions of x_1. If we replace $(x_1,x_2)h_2 = 0$, x_1 for each of its solutions of the form of (76) then we obtain a set of equations in x_2 whose solutions can be obtained in similar manner. Then we proceed analogously fro solving x_3 from $(x_1,x_2,x_3)h_3 = 0$ for each of the possible solutions of x_1 and x_2, and so on until all the indeterminates are solved. This completes the description of the algorithm for solving $(x)F = 0$ over $\mathcal{B}$. We note that the algorithm proposed here, is just the construction of a set of chains of the congruence Θ defined in Theorem 1 for the special case in which the algebra $\langle Q,\Phi\rangle$ is $\mathcal{B}$.

We conclude this digression, by mentioning that an algorithm, called binary search tree (see Hadley [24]) developed for listing solutions of integer-programming problems could be easily adapted for our case to make a systematic (i.e., irredundant) listing of all the solutions of $(x)F = 0$ obtained by the recursive procedure described above.

Now we show that the algorithm for solving $(x)F = 0$ over $\mathcal{B} = \{\{0,1\}, (\vee, \cdot, 1,0,1)\}$ $F : \mathcal{B}^n \to \mathcal{B}$ discussed above can be used, in principle[1], for solving algebraic equations over $\langle Q,\Phi\rangle$, the canonical

[1]We will clarify this qualification later on.

algebra of a given DHA A, for the form

$$(x_z + \alpha_{i,1},\ldots,x_z + \alpha_i m_i) f^{(z)v_j} = (z)(C)\text{proj}_2^{a}$$

$$(z)\ v_j \Psi = g_i$$

$$(z) g_i = (\alpha_{i,1},\ldots,\alpha_i,m_i)$$

First, the following trivial lemma.

Lemma 4. Let $\langle Q,\Phi\rangle$ with $|Q|$ finite, be a given universal algebra with label set G. For any $\gamma \in G$, the operation $f^{\gamma} : Q^{n_\gamma} \to Q$ can be represented faithfully by a finite set of Boolean functions $\{F_\gamma^{\ell} : B^{m_\gamma} \to B,\ \ell=1,\ldots,\mu\}$ for suitably chosen numbers m_γ and μ.

Proof:

Choose m_γ and μ according to

$$\left.\begin{aligned} 2^{m_\gamma - 1} &\le |Q|^{n_\gamma} \le 2^{m_\gamma} \\ 2^{\mu-1} &\le |Q| \le 2^{\mu} \end{aligned}\right\} \qquad (77)$$

Let E_{n_γ} be the n_γ-iterate of the Cantor-pairing function defined in Section 4.2 and let $D_{m_\gamma} : IN \to \hat{B}$ be a binary encoding function such that for any integer n,

$$0 \le n \le 2^{m_\gamma} \qquad (78)$$

---

[1] See (6) or (24) and companion discussion.

gives its binary expansion in m_γ bits (this expansion can be made 1-1 in virtue of (78), but is not onto in general). Similarly, let D_μ be a binary encoding function such that for each $n \in IN$ nD_μ gives its binary expansion in μ bits.

Define $F^\ell_\gamma : \mathcal{B}^{m_\gamma} \to \mathcal{B}$, $\ell=1,\ldots,$ as follows for $(q_1,\ldots,q_{n_\gamma}) \in Q^{n_\gamma}$

$$((q_1,\ldots,q_{n_\gamma})E_{n_\gamma})E_{n_\gamma}D_{m_\gamma})F^\ell_\gamma = (q_1,\ldots,q_{n_\gamma})f^\gamma D_\mu e^\mu_\ell \qquad (79)$$

$$\ell=1,\ldots,\mu$$

where e^μ_ℓ picks the ℓth bit in the expansion $(q_1,\ldots,q_{n_\gamma})f^\gamma$.

Clearly, the left-hand side of (79) is a Boolean function for each ℓ and therefore can be expressed as

$$(x)F^\ell_\gamma = \bigvee_{A \in \mathcal{B}^{m_\gamma}} (A)F^\ell_\gamma x^A \qquad (80)$$

$$\ell=1,\ldots,\mu$$

where $x = \{x_1,\ldots,x_{m_\gamma}\}$ is a set of indeterminates over $\mathcal{B}$.

Corollary: If $|\Phi|$ in Lemma 4 is finite then $\langle Q,\Phi\rangle$ can be represented faithfully by a set of Boolean functions over B.

Proof:

Choose

$$2k+1 = n = \max_{\gamma \in G^\gamma} n$$

$$m = \max_{\gamma \in G^\gamma} m \qquad (81)$$

and proceed as in the lemma for each $\gamma \in G$ with m instead of m_γ.

Finally, we replace in each algebraic system $j \in J'$ of the form of (24) (over $\langle Q,\Phi\rangle$), each equation by a corresponding set of μ equations over $\mathcal{B}$ of the form

$$(x)F^{\ell}_{(z)}v_j = (C)\ \mathrm{proj}_2\ D_{\mu}e^{\mu}_{\ell} \qquad \ell=1,\ldots,\mu \qquad \forall\ z \in S \tag{82}$$

with $F^{\ell}_{(z)}v_j$ as in (80).

Then we convert the resulting system into an equation of the form $(x)F = 0$ over $\mathcal{B}$ using Theorem 6 (iii), and proceed with the algorithm for solving these equations discussed above. Let $x = [x_1,\ldots,x_m] = \beta \in \mathcal{B}^n$ be a solution; then the corresponding solution over x^o is determined by

$$x^o = x\ D^{-1}_m\ E^{-1}_{n_\gamma} \tag{83}$$

We remark that due to the fact that the encoding D is not onto, this solution may not be meaningful over $\langle Q,\Phi\rangle$ but all the meaningful solutions over $\langle Q,\Phi\rangle$ have meaningful solutions of $\mathcal{B}$ because of its 1-1-ness.

We close this section with a final comment. Although, as we mentioned early, the Boolean construction described above is not implementable in present day, commercially available computers dues to its high dimensionality, research on machines with highly parallel logic units (in fact, distributed machines) are under active development at the present (see for an example Foster [10] and Goodwyn [14]). The machine

considered by these authors is a CAPP: Context Addressable Parallel Processor) and there is hope that algorithms such as the one described above will be practically implementable for moderately sized Boolean equations i.e., $|x| \sim 12k$).

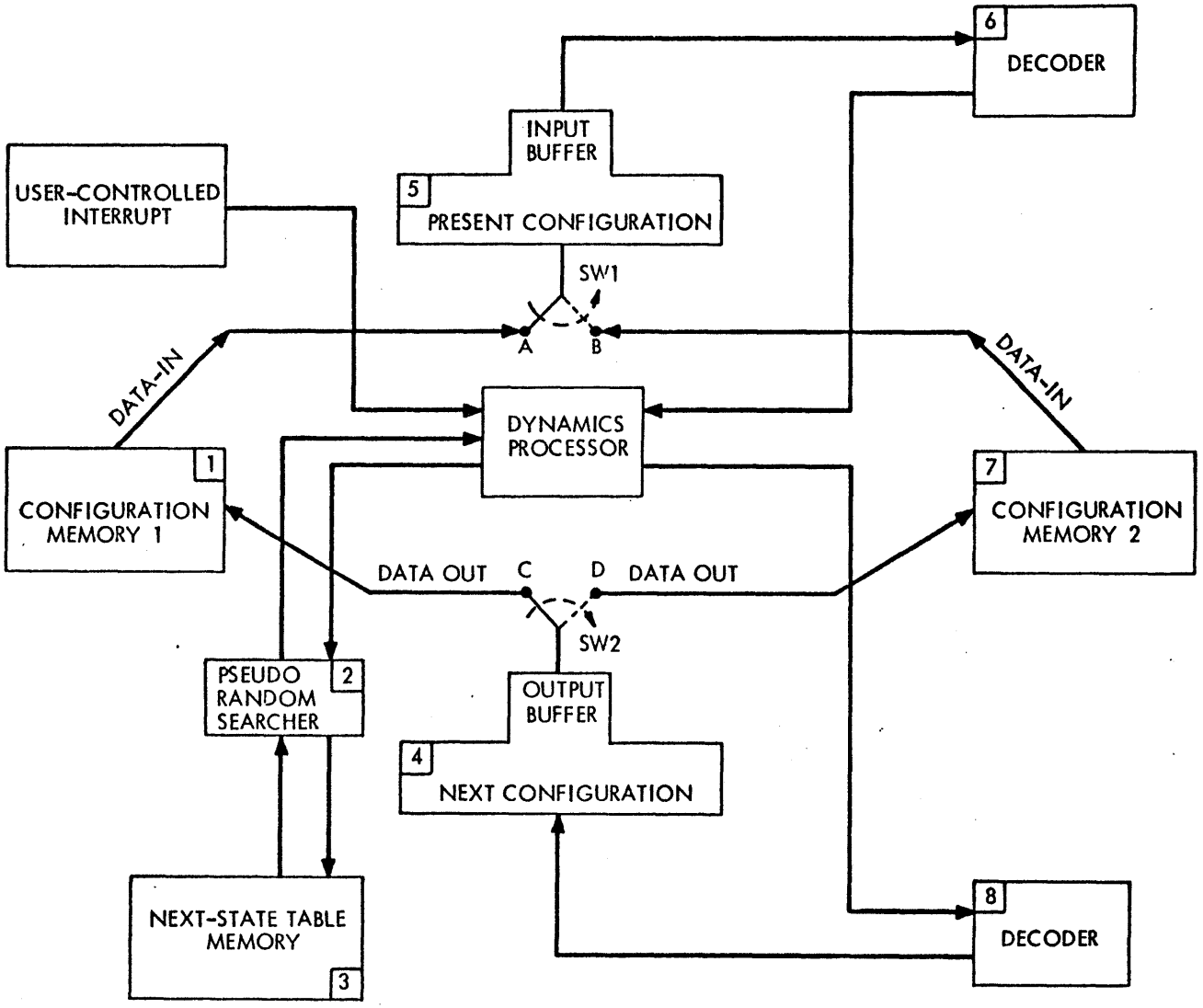
5. SIMULATION STUDY OF EPIGENETIC CONTROL MECHANISMS

5.1 Computer Implementation of DHA's

In this section we describe a computer program called DDHA, for the dynamic simulation of DHA's. Although we concentrate on the implementation of DHA's with epigenetic control applications in mind (see Section 5.3). The program, whose general characteristics were introduced in Section 2.3, (see Figure 2.3-2 and companion discussion) implements the one-step configuration transition map CTM L of a given DHA A. DDHA operates locally on the representation of a configuration (present time configurations) on a fast-access memory and produces the next (in time) configuration which is stored in a second fast-access memory. The latter then takes the place of present configuration and the process is repeated as illustrated in Figure 2.3-2.

A more or less detailed schematic of the dynamic operation of DDHA and its functional components is shown in Figure 1. We proceed next to describe each functional block of Figure 1 with respect to two aspects: construction (programming, storage, etc.) of the block and operational characteristics.

We start with block 3; NEXT-STATE TABLE MEMORY. This block, as its name indicates, is a storage of the next structure function H and the next intensity functions $f^{\gamma} \in \Phi$ of A. Of course, by storage of these functions we mean the storage of tables of their graphs.

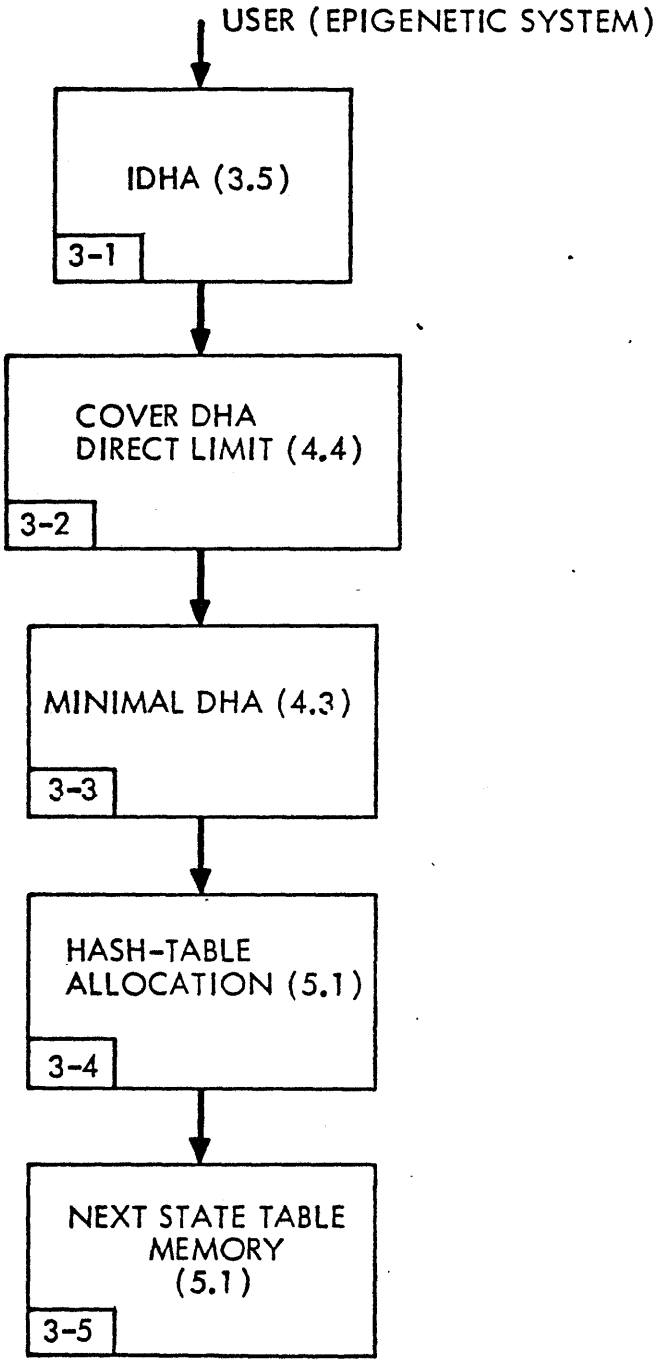


General Organization of Program-System DDHA
Figure 1

Recall from Section 3.5, that the outcome of the program IDHA is precisely a set of those tables for a subset of types of the model of the genetic apparatus developed in Section 3.3. These tables, formed by the arrays TYPES, TYPEI, and GN are not next-state transition tables for a DHA; so, our first task is to convert them into that form. This is done by the program (actually an overlay of programs) whose block diagram is shown in Figure 2. The number and/or captions in parenthesis indicate the section in which the particular program or procedure is discussed. Thus, block 3-1 represents IDHA the conversational program for constructing the model from descriptive interaction with the user.

3-2 represents the construction of the direct limit cover DHA from the set types obtained from IDHA. Recall that this gives us a DHA A, which simulates each of these types. In block 3-3 , we obtain the minimal DHA A_K using the algorithm shown in block diagram form in Figure 4.3-1 and then, we come to an operation, called HASH TABLE ALLOCATION which is extremely important in the fact that its outcome determines in considerable part the time retrieval efficiency of DDHA, and also, as mentioned in Section 4.5, the efficiency of the algorithm for solving algebraic systems of equations for the three types of epigenetic control problems.

By Hash table allocation, we mean (see Knuth [ ] or Aho and Ullman [ ]) a mechanism for storing efficiently a table of entries. This mechanism is usually a function h called a hashing function, a hash table, and a data storage table. These items are illustrated in the diagram of Fig-



(Numbers in parenthesis correspond to sections in which the corresponding procedures are described)

Program Overlay for the Construction of
Next State Table

Figure 2

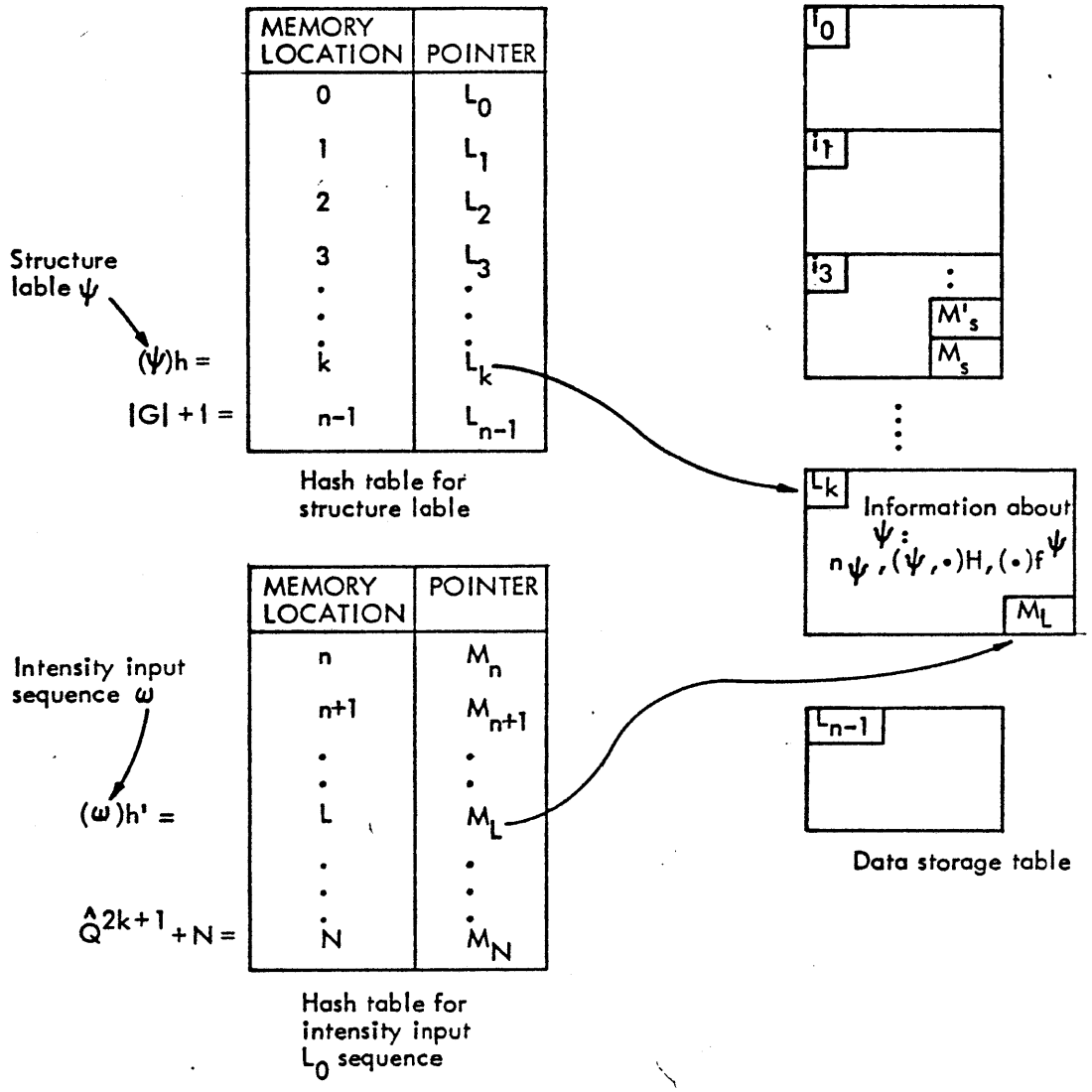
ure 3. Notice that because of the particular structure of our transition functions two hashing functions and two hash tables are required. Next, we briefly explain the general principles underlying the operation of the hashing mechanism.

Each entry in the hash table consists of two items; and identifier (sometimes called name) and a pointer. For our purposes we need two hash tables (see Figure 3). In one of them, the identifier for each entry is the label of one and only one structure γ of the DHA under implementation, and the pointer is the address of a memory location which heads (i.e., is the fist memory location) of a memory area in which the tables for $(\cdot)f^{\gamma}$ and $(\gamma,\cdot)H$ are stored. In the other, the identifier for each entry is a unique sequence of intensities of $\hat{Q}^{2k+1}$, ω, and the pointer is a memory location in each of the memory areas in the data storage table determined by the first Hash table (see Figure 3).

Both hashing functions h and h' in Figure 3 are actually sets of functions $h = (h_o,\ldots,h_m)$, $h' = (h'_o,\ldots,h'_{m'})$ respectively, where

$$\left.\begin{array}{l} h_i : G \to \{0,1,\ldots,n-1\} \quad i=0,\ldots,m \\ \\ h'_j : \hat{Q}^{2k+1} \to \{n,n+1,\ldots,N\} \quad j=0,\ldots,m' \end{array}\right\} \qquad (1)$$

The special characteristics of the functions h and h' will be described in a moment, but before that we want to explain our algorithm for



Next State Table Memory

Figure 3

generating the Hash table. We refer here to h_o of h as the primary hashing function[1] of h, and to h'_o of h', as the primary hashing function of h'.

Recall from Figure 2 that the input to the algorithm for hash-table allocation is the minimal DHA A_K being implemented. Thus, initially we have divided a region of memory into n parts that we call volumes. Each volume is identified by a unique pointer in the Hash table; for structure labels (see Figure 3). Similarly, each volume is divided into N-n parts (not of equal size in general) called files, each of which is identified by a unique pointer in the Hash-table for intensity input sequences.

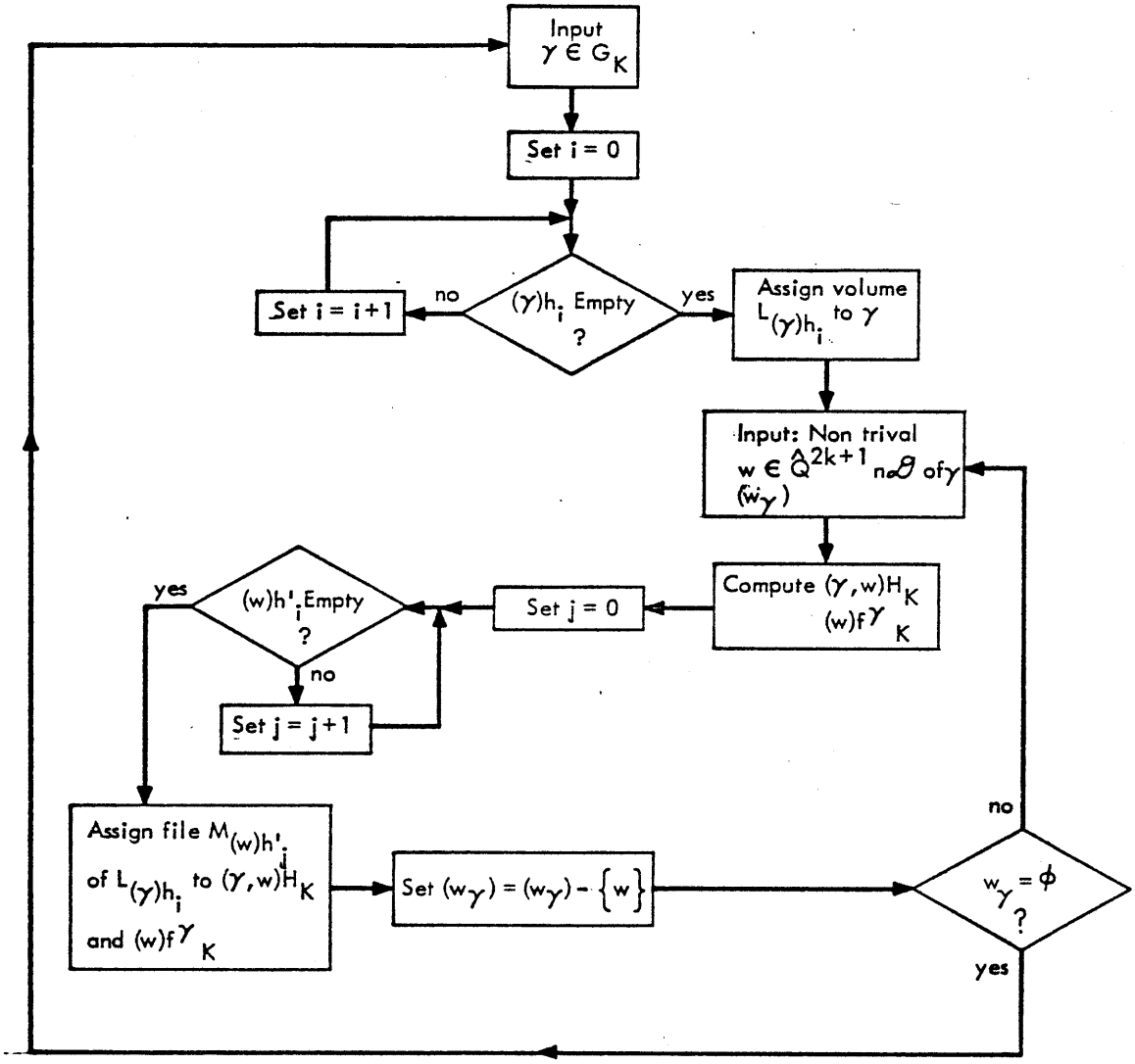
The algorithm works as follows. Initially all the entries in the memory location-column of both hash tables are empty[2]. We pick a $\gamma \in G_K$ the structure label set of A_K, and compute $(\gamma)h_o$. This gives us a location in the Hash table for labels, for the label (γ). We proceed in this fashion for a new label γ' and so on, until for some γ'', $(\gamma'')h_o = k$ and k has already been assigned. In this case, we compute $(\gamma'')h_1$ which gives us a new location k'. If this in turn has already been assigned, then, we compute $(\gamma'')h_2$ and so on until, for some h_i, $i \leq m$ $(\gamma'')h_i$ gives us an empty location.

Once a pointer and thus a volume for a structure γ has been obtained

---

[1]This terminology is standard in the literature.

[2]i.e., it has -0 (minus zero) in each of the entries.



Algorithm for Computation of Hash Table and Data Storage Table

Figure 4

we compute $(\omega,\gamma)H_K$ and $(\omega)f_K^\gamma$ for each ω for which these computations are not trivial[1], and store the results in the file determined by the pointer in the hash table of intensity sequences corresponding to $(\omega)h_j$, $j \leq m'$.

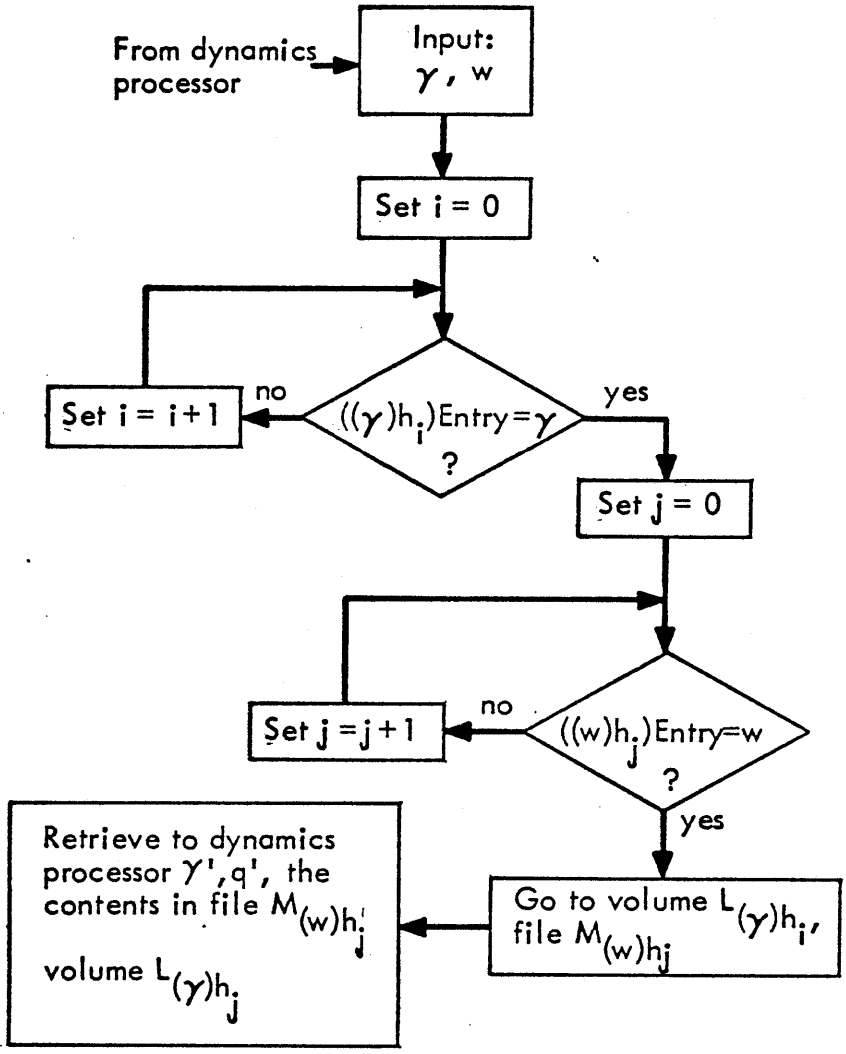
We proceed in this manner until all the state transitions have been allocated to the files of the data storage table. The algorithm described above is shown in block diagram form in Figure 4.

Now we describe how this storage allocation operates during the dynamic simulation of a given DHA. Suppose the program DDHA is engaged in the computation of the next state at some point z; this is done by the dynamics processor (box 9 in Figure 1). Let γ be the present intensity and ω the input sequence of the element at z. Then, the processor sends a request (i.e., sends γ and ω) to the pseudo-random searcher[2] (box 2 in Figure 1) this searcher is a routine that computes $(\gamma)h_o$. If the entry in the hash table for structure labels corresponding to $(\gamma)h_o$ is γ, it proceeds to compute $(\omega)h_o'$ again, if the entry in the hash table corresponding to $(\omega)h_o'$ is ω, we go to the respective file and retrieve to the dynamics processor the numbers $(\gamma,\omega)H_K$ and $(\omega)f_K^\gamma$. If, on the other hand, $(\gamma)h_o = k$ and the entry in k is not γ we proceed to compute $(\gamma)h_1 = k'$ and see if the entry k' is γ; if it is, we proceed to

---

[1]Recall from 3.3 that only a few $\omega \in \hat{Q}^{2k+1}n\mathcal{D}_o f^\gamma$ have operational meaning, all the others are either completion intensities (see Section 4.2) or the dormant intensity.

[2]This denomination will become clear once we describe the nature of the functions h and h', respectively.



Pseudo Random Searcher Algorithm

Figure 5

check the corresponding inquest in $(\omega)h'_o$; if $(\omega)h'_o = \ell$ and the entry ℓ is ω we proceed to retrieve $(\gamma,\omega)H$ and $(\omega)f^{\gamma}$; if it is not then we compute $(\omega)h'_j$ for j=2,3,..., until for some j we find $(\omega)h'_j = k''$ and the entry in k" is ω. This procedure is summarized in the block diagram of Figure 5.

We now conclude our discussion of the next state table allocation in the program with an explicit description of the hashing functions h and h'.

For i=1,...,m

$$(\gamma)h_i = ((\gamma)h_o + r_i) \bmod n \text{ with } m=n-1,\ n = |G_K| \quad \forall\ \gamma \in G_K \tag{2}$$

For j=1,...,m'

$$(\omega)h'_j = ((\omega)h'_o + r'_j) \bmod N \text{ with } m'=N-1,\ N = |\hat{\Omega}^{2k+1}_K| \quad \forall\ \omega \in \hat{\Omega}^{2k+1}_K \tag{3}$$

where r_i i=1,...,m are pseudo random variables $r_i : \Omega \to \{1,...,n-1\}$ and r'_j are random variables $r'_j : \Omega \to \{n,...,N+n-1\}$. In our study both r_i i=1,...,m and r'_j j=1,...,N have uniform distributions.

The random number generators that we use for generating the r_i's and r_j's are listed in Appendix A.3. These generators generate every integer between 1 and n-1 (resp. between n and n+N-1) exactly one in n (resp N) calls. Each time the functions h_i i=1,...,n-1 (resp h'_j j=1,...,N-1) are used the random number generator is reinitialized to the same point. Thus, the sequence $r_1, r_2,...$ is generated each time

h(resp h') is used (either for storage or retrieval).

The functions $(\gamma)h_o$ and $(\omega)h'_o$ are defined as follows:

Let

$$\gamma = \alpha_o 2^o + \alpha_1 2 + \ldots \alpha_k 2^k \tag{4}$$

i.e., the binary expansion of γ. Then

$$(\gamma)h_o = \sum_{s=0}^{\lceil k/2 \rceil} \alpha_s 2^s \tag{5}$$

Similarly, let

$$\omega = \beta_o 2^o + \ldots \beta_\mu 2^\mu \tag{6}$$

then

$$(\omega)h'_o = \sum_{\ell=0}^{\lceil \mu/4 \rceil} \beta_\ell 2^\ell \tag{7}$$

The reasons for these choices is that the right hand sides of (5) and (7) are easily generated (in one step, in fact) with a digital computer and produce meaningful memory allocation numbers (see Figure 3). The reasons for taking $k/2$ and $\mu/4$, respectively is purely empirical based on experimentation with the ADAGE AGT-110 computer used in the implementation discussed in this section. We mention as an indication of the size of the hashing tables, that $|G_K|$ for the lac operon example (see Section 3.5) is equal to 36 and $|\hat{\Omega}^{2k+1}|$ (only the meaningful sequences) is equal to ~280.

We conclude this discussion of the storage of the state transition table with a brief analysis of the efficiency of the proposed storage with respect to a) the expected number of probes k required for inserting a pair $(\gamma,\omega)H_K$ $(\omega)f_K^{\gamma}$ in the data storage table, i.e., expected value of the number ℓ=i+j in the program of Figure 4 when the program exists through the * link after a pair (γ,ω) has been put for storage; and b) the expected retrieval time τ that is, the expected number of probes required to retrieve a pair $(\gamma,\omega)H_K$, $(\omega)f_K^{\gamma}$ once the request has been put by the dynamics processor, that is, the expected value of i+j in the program of Figure 5, when the program exits through the ** link.

In order to evaluate $\ell = E\{i+j\}$ i,j as in program 4, and $\tau = E\{i+j\}$ i.j as in program 5, we need to assign probability distributions to occurences of elements $\gamma \in G_K$ and $\omega \in \hat{Q}_K^{2k+1}|$ (useful ω's only). We have, after an observation of the assignments given by IDHA to the types of the lac operon assigned the following probability distributions to items of G_K and $\hat{Q}_K^{2k+1}$. For any $\gamma \in G_K$

$$P_G(\gamma) = \frac{1/\gamma}{\sum_{\gamma' \in G_K} 1/\gamma'} \qquad (8)$$

that is, γ with lower values have higher probability of occurence (see example of Section 3.5 to convince yourself that this is the case) the conclusion that we will obtain in a moment is the same if we assume the probability of the γ's to occur to be uniform for any useful $\omega \in \hat{Q}_K^{2k+1}$

$$P_{Q}(\omega) = \frac{(\omega)E_{2k+1}}{\sum_{\omega' \in \hat{Q}_K^{2k+1}} (\omega')E_{2k+1}} \tag{9}$$

where E_{2k+1} is the 2k+1-iterate of the Cantor pairing function (see Section 4.2). Finally, let $P_r(r_i)$ i=1,...,n $P_{r'}(r'_j)$ j=1,...,N be the probability distribution of the random numbers in (2) and (3) respectively. Thus, we have that the expected number of probes required to insert a pair $(\gamma,\omega)H_K$, $(\omega)f_K^{\gamma}$ in the storage table when x out of n-1 entries in the hash table of structures are filled and y out of the N-1 entries of the input intensity hash table are filled, is given by

$$E(x,y) = \sum_{m=1}^{x} m \sum_{\gamma \in G_K} P_G(\gamma) \sum_{\mu \in \{1,...,n-1\}} P_r(\mu)$$

$$+ \sum_{s=1}^{y} s \sum_{\omega \in \hat{Q}_K^{2k+1}} P_Q(\omega) \sum_{a \in \{1,...,N-1\}} P_{r'}(a) \tag{10}$$

where in (10) we have used the assumption that r_i's and r'_j's are independent as well as the occurrences of $\gamma \in G_K$ and $\omega \in \hat{Q}_K^{2k+1}$. The worst case would be clearly, when $x\alpha$ is the last γ to be stored and $y\alpha$ is the last γ to be stored, since in that case the table has the maximum number of entries filled and program 4 has to probe with almost all the hashing functions (6) and (7). For this situation and in the example of the lac operon system whose implementation will be discussed at the end

of this section we found $E(x,y) \simeq 4.88641$; that is, approximately 5 steps in the program 4 are required for storing $(\gamma,\omega)H_K$ and $(\omega)f_K^\gamma$. For an intermediate value of x, y we found that $E(x,y) \simeq 3.79952$. In contrast, with linear lexicographic storage, the number of steps required for storing a pair $(\gamma,\omega)H_K$, $(\omega)f_K^\gamma$ equals essentially the sum of the numbers of labels from G_K stored so far plus, the number of ω's in the corresponding $\mathcal{D}_o f^\gamma$. Thus, in the average we need about 18 + 200 = 218 steps for this type of storage in the lac operon system example. Clearly, the random storage algorithm is vastly superior in this context.

Similarly, the number $E(\tau)$ of expected probes required for retrieving a pair $(\gamma,\omega)H_K$, $(\omega)f_K^\gamma$ equals, assumming that any pair in the table is equally likely to be retrieved, to the average number of probes that were required originally for inserting this pair into the table, i.e.,

$$E(\tau) = \frac{1}{n \cdot N} \sum_{\ell=0}^{n-1} \sum_{y=0}^{N-1} E(\ell,y) \qquad (11)$$

For the lac operon system, using (11), $E(\tau) \simeq 2.53$! This, compared with the average retrieval time in the case of linear lexicographic search for this case, which is equal to $N \cdot n/8 \simeq \frac{280 \times 36}{2}$ shows that the random storage (pseudo random storage actually) of the transition table makes the <u>crucial</u> ingredient in making feasible the implementation of a DHA in a digital computer.

We mention that in the actual simulation, carried out for the lac

operon system, which is described at the end of this section, E(T) was found to be larger (6 to 10) than the theoretical value computed with (11). However, this is still vastly superior to the corresponding figure for the linear lexicographic search. A reason for the discrepancy between the actual and the theoretical values for E(T) may be, in part, due to the fact that the random number generators for the r_i's and r_j''s are not really uniform (in fact, they are not even purely random) which was an assumption in the computation of (10) and (11).

We note that the pseudo-random searcher program described above and whose block diagram appears in Figure 5, can be used as the search procedure in the solution of the initial activation control problem described in Section 4.5 and we will illustrate this fact with an example at the end of this section.

Now we describe the Encoder (Box 8 in Figure 1) and Decoder (Box 6 in Figure 1) procedures. In order to do that, we must first describe how the configurations (Boxes 1 and 7 in Figure 1), and how the two buffers (5 and 4 in Figure 1) operate on the contents of the two memories.

In storing a configuration A in either memory configuration we use the following conventions.

1 - A location in either memory is either empty or contains an encoded version of the state[1] of a non-dormant element $(z)T$, allocated at $z = (z_1, z_2, z_3) \in Z^3 = I$.

[1] By state, we mean of course, structure and intensity.

2 - The addresses of the memory configurations of which non-dormant states of elements are allocated are determined by the following rule: for the element allocated at $z = (z_1, z_2, z_3)$, the corresponding address $(z)\alpha$ is given by

$$(z)\alpha = (z_1, z_2, z_3)E_3 \tag{12}$$

where E_3 is the 3-iterate of the Cantor pairing function (see Section 4.2).

3 - If the element allocated to z, with address (z)a, is in state (γ,q), at present time, then the content of (z)a is the number

$$((z)\alpha) \text{ content} = (\gamma,q)E_2 \tag{13}$$

where E_2 is the Cantor pairing function.

4 - In each memory configuration, the non-empty addresses are grouped into sets of 208 addresses[1] called files.

With these 4 conventions, we are now ready to discuss the Encoder and Decoder procedures. Suppose that at time t of the simulation, configuration memory 1 (Box 1) contains the present configuration $(C_t\ (C_t)\text{Supp}_o)$ encoded as indicated above. When the dynamics processor generates an interrupt, a file (208 numbers) from memory 1 are brought into the input buffer-present configuration (Box 5). The first task of the decoder is to compute v_t and ω_t, i.e., the present structure and

[1]The reason for that is that the actual implementation was carried out in an ADAGE AGT-110 computer in which the disk buffers, which are the buffers that we use in our program, are 208 words long.

and present intensity corresponding to each of these 208 elements. This computation is summarized by the following:

Let z_i, i=1,...,208 be the coordinates of the elements in the buffer; then $\alpha_i = (z_i)\alpha$ i=1,...,208 are the corresponding addresses in memory configuration 1.

Hence, we obtain the coordinate $z^i = (z_1^i, z_2^i, z_3^i)$ i=1,...,208 from its address α_i as,

$$z^i = (z_1^i, z_2^i, z_3^i) = (\alpha_i)\alpha^{-1} = (\alpha_i)E_3^{-1} \quad i=1,...,208 \qquad (14)^1$$

$$\text{Let } (\alpha_i) \text{ content} = \beta_i \quad i=1,...,208 \qquad (15)$$

then,

$$\left.\begin{aligned} (z^i)v_t &= (\beta_i)E_2^{-1}\ \text{proj}_1 \\ \text{and} \qquad & \qquad i=1,...,208 \\ (z^i)\omega_t &= (\beta_i)E_2^{-1}\ \text{proj}_2 \end{aligned}\right\} \qquad (16)$$

Combining (14) and (15) we obtain an expression that summarizes the computation of the procedure decoder:

$$\left.\begin{aligned} (\alpha_i)E_3^{-1}v_t &= (\alpha_i)\ \text{content}\ E_2^{-1}\ \text{proj}_1 \\ & \qquad i=1,...,208 \\ (\alpha_i)E_3^{-1}\omega_t &= (\alpha_i)\ \text{content}\ E_2^{-1}\ \text{proj}_2 \end{aligned}\right\} \qquad (17)$$

---

[1] Recall that $E_s : IN^s \to N$ is invertible (see Section 4.2).

Similarly, assume now that the output buffer-next configuration (Box 4) contains 208 structure-intensity pairs $(z^i v_t, z^i \omega_t)$ corresponding to elements at coordinates z^i i=1,...,208 then, we compute the corresponding addresses of memory locations to which these elements must be allocated in memory configuration 2, (which is being loaded with the next in-time configuration) according to

$$(z^i)\alpha = (z_1^i, z_2^i, z_3^i)E_3 \tag{18}$$

and

$$(z^i)\alpha \text{ content} = (z^i v_t, z^i \omega_t)E_2 \qquad i=1,...,208 \tag{19}$$

Procedure encoder is realized in our program DDHA, by a subroutine named Folding, and procedure decoder is realized by a subroutine called Unfolding, which are listed in Appendix A.3.

Now we describe the dynamics processor procedure (Box 9) in Figure 1). This procedure has two tasks

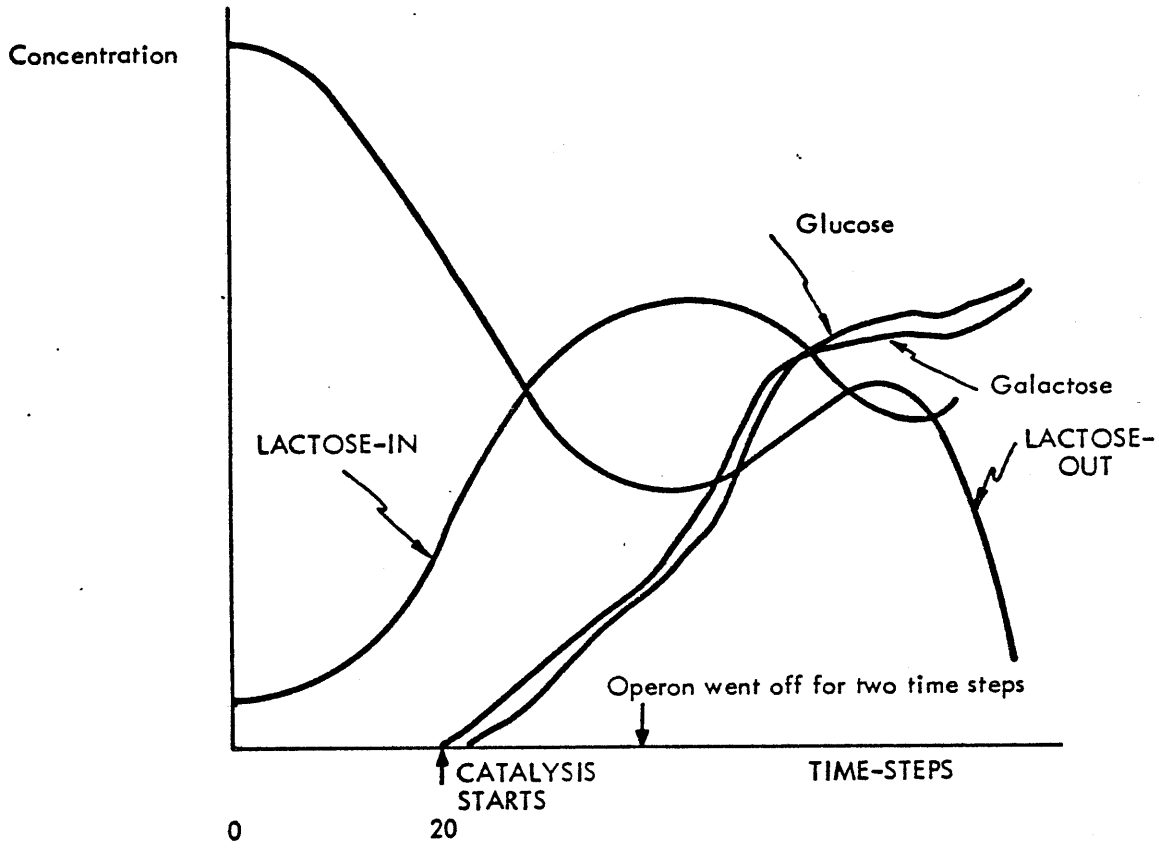
a) Control the actions of the other blocks in the program (DDHA); and b) Carries out the computation of next states for the elements allocated in the input buffer.

We describe task b) first:

Assume, as above, that the present time configuration is allocated in memory configuration 1; the other case is perfectly symmetric.

We have in the input buffer 208 pairs of states

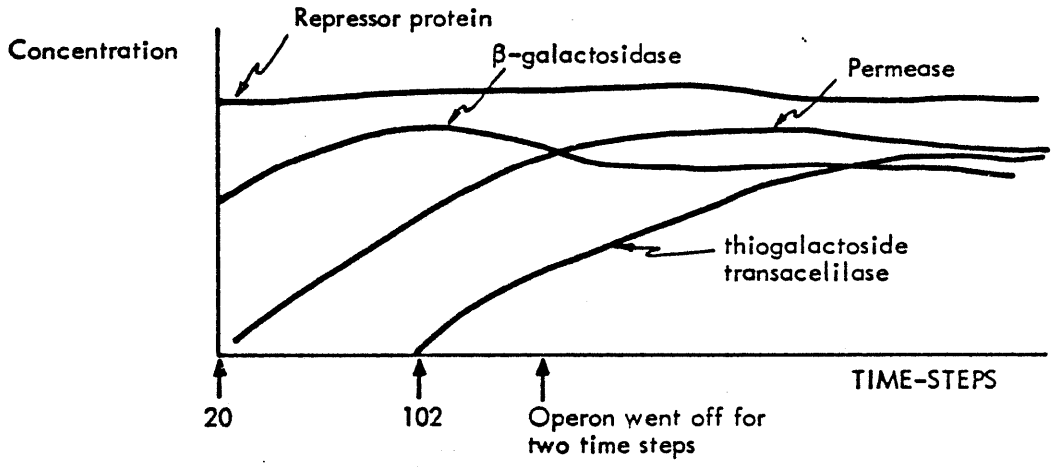
$$(z^i v_t, z^i \omega_t) \quad i=1,...,208$$



(time scale is logarithmic)

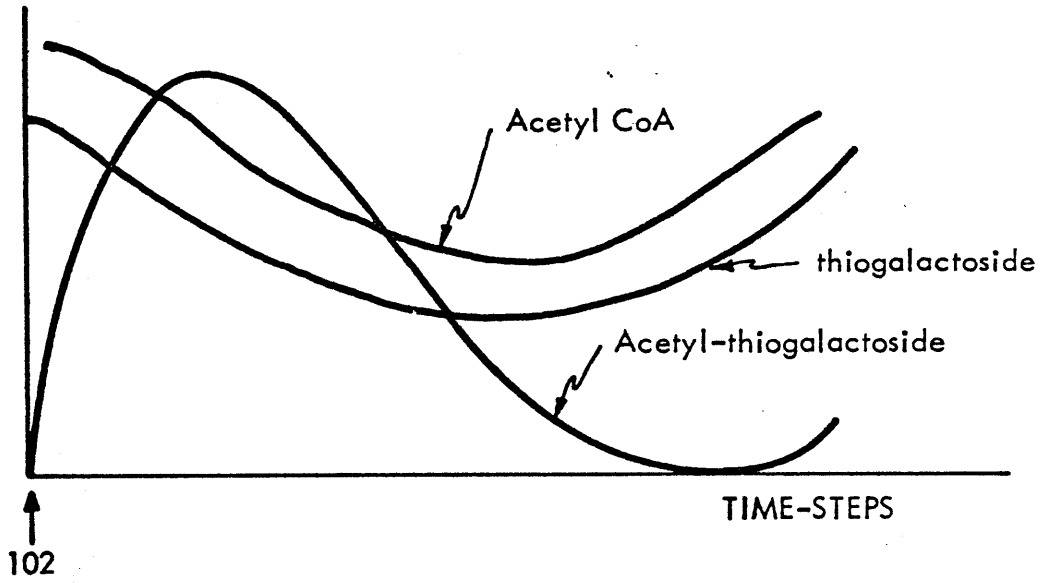
Reaction Systems (1,2) (see Figure 3.5-5)
(1,3)

Figure 6



Protein Concentrations

Figure 7



Reaction System (1,4) (see Figure 3.5-5)

Figure 8

obtained from the decoding operation (17). The dynamics processor proceeds then as follows:

1 - Set i-1

2 - Compute $z^i v_t \Psi = g_j$ where Ψ is the input selector function of the DHA $_K$ under simulation[1]

3 - Compute $(z^i)g_j = (z^i + \alpha_{j,1},\ldots,z^i + \alpha_{j,m_j})$

4 - If $(z^i + \alpha_{j,1},\ldots,z^i + \alpha_{j,m_j}) \subseteq \{z^i,\ i=1,\ldots,208\}$ Go to 6; else go to step 5

5 - Bring to input buffer the decoded contents of all elements at points $z^i + \alpha_{j,k} \notin \{z^i,\ i=1,\ldots,208\}$.

Step 5 has the objective of bringing the additional nearest neighboring states of the element at z^i required for the computation of its next state.

6 - Set $\gamma = z^i v_t$, $\omega = ((z^i + \alpha_{j,1})\omega_t,\ldots,(z^i + \alpha_{j,m_j})\omega_t)$

7 - Call pseudo random searcher (Box 12).

We recall that this procedure, described earlier, retrieves $\gamma' = (\gamma,\omega)H_K$, $q' = (\omega)f_K^{\gamma}$.

8 - Call Encoder (Box 8). This procedure computes:

$(z^i)\alpha = (z_1^i, z_2^i, z_3^i)E_3$, $(z^i)\alpha$ content $= (\gamma',\omega')E_2$

---

[1]We could have randomized the table of $\Psi : G_K \to N_K$, but since the number of different neighborhood functions is fairly small (~10) we found this to be unnecessary.

9 - Send $(z^i)\alpha$, $(z^i)\alpha$ content to output buffer

10 - Set $i \doteq i+1$

11 - If $i < 208$ go to 2; else go to 12

12 - Bring a new set of 208 memory contents from memory configuration 1 to input buffer

13 - Send the 208 set of computed next-states from output buffer to encoder - to memory configuration 2

14 - Set $t = t+1$ (t is time)

15 - If $t > T_H - 1$ Halt, else go to 16

16 - Are all the contents in input buffer empty. If yes set SW1 to B, SW2 to D[1] (i.e., now memory configuration 2 contains the present configuration and we will load memory configuration 1 with the next state

17 - Set memory configuration 1 ← memory configuration 2
memory configuration 2 ← memory configuration 1

18 - Go to 1.

Task a) consists of steps 14, 15 for timing of the simulation and steps 16, 17 for switching memory configurations. In addition, it provides with a mechanism for stopping the simulation at any time step (user controlled interrupt).

[1]SW1 and SW2 are the switches shown in Figure 1.

This completes the description of the simulation program DDHA. Now we illustrate its operation with an example of a DHA simulating the dynamics of the lac operon system described in Section 3.5, and whose state assignment was carried out there (by program IDHA). The initial configuration for the reduced DHA A_K is listed in Table 1. The time step is equivalent to 30 sec in the real process, the quantization interval corresponds to 1 pinol/ml. T_H was taken equal to 600 steps (~1800 sec.), the simulation time was about 20 minutes; that is, an average of 2 sec/step.

In Figures 6, 7 and 8 we show the corresponding time evolution of the molecules of the system (see Section 3.5). We used a 7 order polynomial curve fitting to interpolate between sample levels. (The plots are generated by a special graphics program called Screen (see Kohn and Johnson [ ]).)

In Table 1, the structures and intensities of the different elements are the ones assigned by IDHA (see Section 3.5, Table 3.5-1). The initial configuration displayed in Table 1 is not the initial configuration of the minimal DHA A_K, but of the original DHA (constructed by IDHA; see Section 3.5). We put this configuration instead of the actual one used in the simulation in order to correlate the assigned elements with the physical elements they represent.

In Table 1, we have listed only a few of the elements representing lactose-out molecules (structure 33, intensity 49). In the actual initial configuration we have assigned 300 of these molecules. The operon

Table 1

Initial Configuration for Simulation

of Lac-Operon System

DDHA INIITIAL CONFIGURATION

POSITION			STRUCTURE	INTENSITY	
Z1	Z2	Z3			ELEMENT
1	0	0	7	36	OPERON
2	0	0	0	0	CLOCK
1	-1	0	12	0	MRNA POL.
2	-1	0	16	0	MRNA
2	-1	1	20	1	RIBOSOME
3	-1	1	20	1	RIBOSOME
4	-1	1	20	1	RIBOSOME
5	-1	1	20	1	RIBOSOME
6	-1	1	20	1	RIBOSOME
2	-1	2	44	0	PROTEIN
1	1	0	51	0	TFM
1	2	0	29	48	REACT.SYST 3
1	3	0	29	48	REACT.SYST 3
1	4	0	29	48	REACT.SYST 3
1	5	0	29	48	REACT.SYST 3
2	2	0	33	49	REACT.SYST 3
3	2	0	33	49	REACT.SYST 3
4	2	0	33	49	REACT.SYST 3
5	2	0	33	49	REACT.SYST 3
5	3	0	33	49	REACT.SYST 3
5	4	0	33	49	REACT.SYST 3
5	5	0	33	49	REACT.SYST 3
5	6	0	33	49	REACT.SYST 3
5	7	0	33	49	REACT.SYST 3
5	8	0	33	49	REACT.SYST 3
5	9	0	33	49	REACT.SYST 3
5	10	0	33	49	REACT.SYST 3
6	1	0	38	52	REACT.SYST 4
6	2	0	38	52	REACT.SYST 4
6	3	0	38	52	REACT.SYST 4
6	4	0	38	52	REACT.SYST 4
6	5	0	38	52	REACT.SYST 4
40	1	0	27	56	STOP
-20	1	0	27	56	STOP
1	1	1	28	57	REG

is assumed to be initially in stand/by and we have put a few elements representing lactose-in molecules (structure 29 intensity 48) to guarantee that the TFM element will be induced.

The region [1] of operation of the simulation in the informational space is determined by the stop elements (structure 27, intensity 56). This region is determined by

$$\begin{aligned} -20 \leq z_1 \leq 40 \\ -2 \leq z_2 \leq 300 \\ 0 \leq z_3 \leq 2 \end{aligned} \qquad (20)$$

A program for allocating the initial configuration to the memory configuration 1 in coded form, has been designed. This program, called INIT, is listed in Appendix A.3.

The graphs of the simulation[2] are constructed as follows:

Let C_t be the configuration of the DHA A_K at time t, we compute the function $h : Q \to IN$ as follows

$$qh = \text{Number of } z\text{'s} \in (C_t)\text{Supp}_o, \text{ in intensity } q \qquad (21)$$

in the catalysis region (see Section 3.3).

---

[1] See Section 4.5 and Section 3.3.

[2] Figures 6, 7 and 8.

Thus, since we have assumed (see Section 2.1) that the intra-cellular volume is constant, qh is proportional to the concentration of the molecule (either protein or metabolite) q in the cell at time t.

Figure 6 is constructed by computing, at each time step, qh i=1,2,3 for the reactants and products of reaction systems (1,2) and (1,3) (see Section 3.5). Initially, we have constructed our initial configuration so that there is a large concentration of lactose-out elements and a relatively low concentration of lactose-in elements.

When the operon is activated, producing synthesis of MRNA's, and consequently synthesis of the proteins of the operon are accomplished, the elements corresponding to lactose-out molecules disappear as discussed in Section 3.3 and the number of elements corresponding to lactose-in molecules increase (due to the catalytic action of the permease). Simultaneously, the appearance of protein elements representing z-galactosides causes the reaction system to operate with the consequent appearance of elements representing glucose and galactose.

It is interesting to notice that during the simulation, the operon went off for two time steps; an examination of the corresponding two configurations revealed that this was due to the fact that locally there were no lactose-in elements interacting[1] with the TFM, this caused a decrease in the number of MRNA elements (recall that the MRNA elements

[1] i.e., inducing transcription.

are destroyed after the translation of 3 set of proteins; see Sections 3.3 and 3.5) and consequently, a decrease in the number of elements representing proteins.

This decrease is clearly observable in the concentration time-course for the proteins displayed in Figure 7. These simulation results show that the transcriptional control is very effective in sensing environmental changes such as, in this case, the local decrease of control metabolite elements.

Figure 8 shows the behavior of the concentrations for the REACTION system (1,4). The initial number of acetyl C_oA elements allocated was 50. The reversibility of this reaction caused, towards the end of the simulation, a complete disappearance of the elements representing acetylthiogalactoside. The significance of this phenomenon is not clear. We note however, that the correlation between the amount of acetylthiogalactoside elements and acetyl C_oA and thiogalactoside elements (the reactants in (1,4)) obtained from the simulation is consistent.

We conclude this section with some observations about a computation of the initial configuration, using the initial activation process procedure developed in Section 4.5, assuming as given the configuration C_{400} obtained in the simulation described above.

The computation time for the procedure was 18 minutes, the initial configuration obtained is shown in Table 2. The time for the process to reach an initial configuration was equal to 286 steps; that is, from our control algorithm we obtained an initial configuration that reaches

Table 2

Initial Configuration, Result of Initial Activation Process Control Procedure

DDHA INIITIAL CONFIGURATION

POSITION			STRUCTURE	INTENSITY
Z1	Z2	Z3		
1	0	0	7	36
2	0	0	0	10
1	-1	0	12	11
2	-1	0	16	0
2	-1	1	21	1
3	-1	1	20	1
4	-1	1	22	2
5	-1	1	20	1
6	-1	1	20	1
2	-1	2	44	0
1	1	0	51	0
1	2	0	29	48
1	3	0	29	48
1	4	0	29	48
1	5	0	29	48
2	2	0	33	49
3	2	0	33	49
4	2	0	33	49
5	2	0	33	49
5	3	0	33	49
5	4	0	33	49
5	5	0	33	49
5	6	0	33	49
5	7	0	33	49
5	8	0	33	49
5	9	0	33	49
5	10	0	33	49
6	1	0	38	52
6	2	0	38	52
6	3	0	38	52
6	4	0	38	52
6	5	0	38	52
40	1	0	27	56
-20	1	0	27	56
1	1	1	28	57

the configuration C_{400} of the simulation in 286 steps.

A comparison between Tables 1 and 2 shows that the corresponding elements are either in identical states or equivalent ones; for instance, the ribosome element in the simulation at (2,-1,1) is in state (20,0) while, that of the control algorithm is in state (21,1) (i.e., the former has not synthesized any proteins while the latter has synthesized one battery corresponding to one MRNA; see Section 3.3).

The elements representing reaction systems in the catalysis region are in identical states while those in the reactants and products region are in the neutral intensities in the simulation, while those obtained in the algorithm are in intensity, ready for react. Despite these differences, it is interesting to notice that the control algorithm produces an initial configuration which is feasible from the physical point of view. We will come back to this point in the next section where we discuss an algorithm for identifying structures from experimental data when some of the structural characteristics of the system, such as type of TFM of one or more operons of the system is not known.

5.2 Structural Identification of Epigenetic Processes in the Context of DHA's

In this section we develop, on the basis of simulation procedure (DDHA) described in the last section, a procedure for identifying structural characteristics of the elements of an epigenetic control process given two items:

1 - Experimental data of kinetic nature of the characteristics described in Section 2.3[1] (i.e., concentration-time courses of some or all of the chemical elements[2] and/or concentration time courses of some or all of the protein elements participating in the process.

2 - Some partial knowledge about the structural nonobservable characteristics of the elements of the process (e.g., the type of transcriptional feedback mechanism of each of the operons participating in the process, the type of the reaction systems, etc. ...).

We illustrate the characteristics of the identification procedure with the lac operon system described in Section 3.5.

Our strategy in developing the identification procedure, is as follows. First, we transform the given kinetic data into a form that is suitable for its interpretation as observed output of a DHA representing the process (see Section 2.3 and Equation 5.1-21 for a discussion of this transformation). Second, we introduce the concept of schemata of DHA's associated with the process under study as a set of partially specified DHA's sharing some common features to be determined in each specific case, and third, we define a fitness criterion which allows us to choose among the elements of the schemata, a DHA whose characteristics are most likely[3] to correspond to a representation of

---

[1]Figures 2.3.3 and 2.3.4.

[2]By chemical elements we mean elements of the transformation subclass (see Section 3.2).

[3]In a sense to be discussed later.

the process under study.

We start with a description of the usual way in which kinetic data is presented and then we describe a procedure for transforming this data into a form suitable for our purposes.

Figure 1 illustrates a typical presentation of kinetic experimental data (taken from Gutfreund [1]) over a cell population for the reaction system (1,2) (see Section 3.5-1), and normalized of the experimentally measured volume occupied by the population so that it corresponds to a "typical" average cell in the population.

The points encircled represent actual (normalized) measurements, while the solid lines correspond to a 5-order polynomial interpretation under a least squares criterion.

The first important data transformation we perform is time discretization; an estimate of the appropriate time interval[1] for discretization can be obtained by the following procedure.

Recall that the reaction system (1,2) represents the reaction $C(1,2) \overset{PR(1,2)}{\rightleftarrows} A(1,2) + B(1,2)$ where C(1,2), A(1,2), B(1,2) represent lactose, glucose and galactose respectively and PR(1,2) is β-galactoside.

We can write a kinetic model which reproduces the experimental (interpolated) data using a Michaelis-Menton formalism (see Bartholomay [2]). This kinetic model is of the form:

---

[1] See Section 2.3.

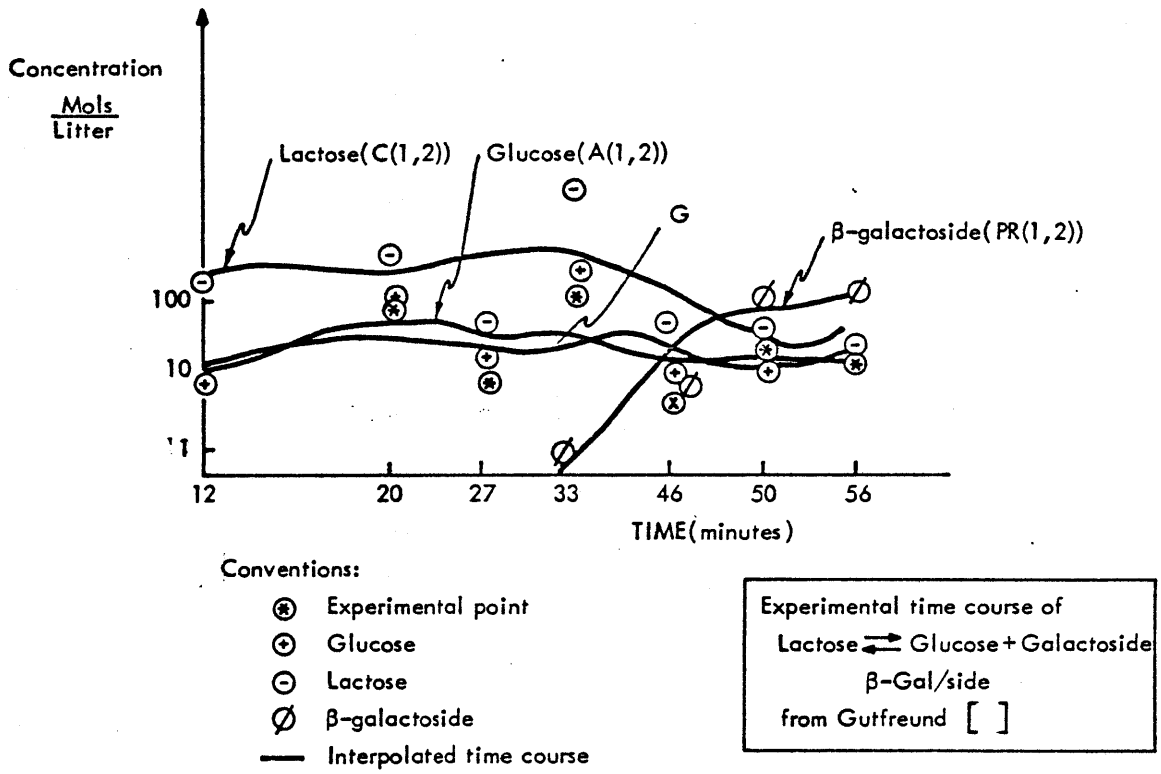

Concentration
Mols
Litter
Lactose(C(1,2))
Glucose(A(1,2))
G
β-galactoside(PR(1,2))
100
10
1
12
20
27
33
46
50
56
TIME(minutes)
Conventions:
Experimental point
Glucose
Lactose
β-galactoside
Interpolated time course
Experimental time course of
Lactose ⇌ Glucose + Galactoside
β-Gal/side
from Gutfreund [ ]

Figure 1

$$\frac{d^2[A(1,2)]}{dt^2} + \frac{d}{dt}[A(1,2)][k_1[C_{1,2}] + k_{-1} + k_2] = k_1 k_2 [C(1,2)] \quad [PR(1,2)]_o \tag{1}$$

with a similar equation for [B(1,2)] where [A] means concentration of species A.

The assumptions in the derivation of (1) are as follows:

a) There is an excess of concentration of the reactant C(1,2), that is, the reaction tends to proceed in the forward direction and the concentration [C(1,2)] does not vary "significantly" over the interval of time under study.

b) The total concentration of enzyme [PR(1,2)] remains constant over the interval of consideration.

k_1, k_{-1}, k_2 are <u>reaction constants</u> (see Wood et al [5]) of the following representation of the reaction

$$C(1,2) \overset{\leftarrow}{\rightarrow} A(1,2) + B(1,2) \tag{2}$$

$$\overset{PR(1,2)}{PR(1,2) + C(1,2)} \overset{k_1}{\underset{k_{-1}}{\rightleftarrows}} PR(1,2) - C(1,2) \tag{3}$$

$$PR(1,2) - C(1,2) \overset{k_2}{\underset{k_{-2}}{\rightleftarrows}} PR(1,2) + A(1,2) + B(1,2)$$

where PR(1,2) - C(1,2) represents an unstable molecule-complex called the activation complex. In fact, (1) can be derived from (3) by a more

or less elementary mass conservation argument (plus the basic Micaelis-Menten assumption; see Bartholomay [2]).

We note that under assumption a) we must restrict ourselves to the interval from 46 to 56 minutes in the graph of Figure 1. Since in this interval, $[k_1[C_{1,2}] + k_{-1} + k_{-2}]$ constant and $k_1k_2[C_{1,2}]$ $[PR(1,2)]$ constant, Equation (3) can be integrated to give

$$[A_{1,2}] = \frac{k_1k_2[C_{1,2}]}{k[C_{1,2}] + K_{-1} + k_{-2}}\, t + \frac{k_1k_2[C_{1,2}][PR(1,2)]_o}{(k_1[C_{1,2}] + k_{-1} + k_{-2})^2}$$

$$\cdot\ e^{[k_1[C_{1,2}] + k_{-1} + k_{-2}]t} + \text{constant terms} \qquad (4)$$

Using Equation (4), and the experimental (interpolated) data of Figure 1, in the time interval [46, 56] we have an algebraic equation for determining the constants k_1, k_{-1}, k_{-2}. We found a solution for this equation using a least squares criterion-algorithm given in Kohn and Johnson [4], whose results are shown in Table 1 below.

Table 1

Least Square Estimates of Reaction Constants of (3)	Estimate
k_1	10,8 sec^{-1}
k_{-1}	30.4 sec^{-1}
k_2	5.1 sec^{-1}
k_{-2}	11.4 sec^{-1}

From (4) and using the parameters in Table 1 we are ready now to state our discretization criterion. Take as time interval of discretization,

the minimum time interval in [46, 56 min] required for the concentration A[1,2] to increase (or decrease) one mole with respect to a standard volume 10^{-8} lit $\simeq$ cellular volume; that is, we consider as the minimal amount of variation in concentration of A(1,2), the one corresponding to the appearance or disappearance of 1 molecule in the system. This criterion is based on the fact that our model considers changes in concentration corresponding to integer numbers of molecules under the assumption that the intra cellular volume is constant (see Section 2.1 for a discussion of this point).

We note that the criterion given above, for choosing the discretization interval, serves also as a criterion for choosing the quantization interval (see Section 2.3; in particular, Figure 2.34 and companion discussion), as the interval representing the change in concentration due to 1 molecule of A(1,2). We choose A(1,2) (which represents glucose) to carry out our analysis because for this specific example, it is the molecule in the system with the smallest molecular weight.

In synthesis, for a general system, we determine an evolution equation for the molecule with the smallest molecular weight using macrokinetic principles (see Bartholomay [2]), use the experimental data to identify the corresponding reaction constants, and then, determine the discretization interval corresponding to the change in concentration for this molecule of 1 mole with respect to the standard volume of the system (i.e., the cell), and the quantization interval equals 1 mole of the molecule in the system with smallest molecular weight.

We then, on the basis of the determined quantization interval, for the molecule of smallest molecular weight proceed accordingly, to quantize the concentration time-courses of the rest of the molecules in the system i.e., if the concentration of i species at time t is [i], we take as its quantized verions $[i]_{min}$, where $[a]_{min}$ is the integer number of quantization intervals smaller than or equal to [a] .

We mention that for the data of Figure 1, the discretization interval equals 37 seconds. This time is considerably larger than the one used in the simulation example in Section 5.1; however, recall that the data used here is the one corresponding to an "average" cell over a given population so that the data does not correspond to a particular cell, but rather to a smoothed version of the cells in the population.

Now, we pass to define the output equation compatible with quantized and discretized data for our simulation family of DHA's (the schemata).

Let $A = \langle I, \tau \rangle$ be the DHA representing a given epigenetic system. Let $P = \{C_o, C_1, \ldots, C_t, \ldots\}$ be the process under study. Define for $q \in Q$, the intensity set of A the function, $\delta_q^k : Q \to \{0,1\}$ be as follows. For C_t any configuration of P, $\forall z \in I$,

$$((z)C_t proj_2)\delta_q^t = \begin{cases} 1 \text{ if } (z)C_t proj_2 = q \\ 0 \quad \text{otherwise} \end{cases} \tag{5}$$

Define the functions $h_q^t : Q \to IN$ (the natural numbers) for some $q \in Q$, $q \neq 0$ (the dormant intensity), $t \in \{0,1,2,\ldots\}$

$$((z)C_t proj_1)h_q^t = \sum(((z)C_t)proj_2)\delta_q^t \qquad (6)$$

$$z \in (C_t)Supp_o$$

That is, (6) is a formal difinition of the function h introduced in 5.1-21 if we restrict ourselves in (6) to points z in I such that

$$z \in (C_t)Supp_o \cap S_{pr} \qquad (7)$$

where S_{pr} is the products and reactants region of the informational space, (see Section 3.3, Figure 3.3-12 and companion discussion). We call h_q^t the aggregation output of $\mathcal{A}$ with intensity q.

Recall our convention in Section 3.3, in which we use the same symbol to represent both the name of a molecule and its intensity in the reactants and product region. With this convention, h_q^t represents the (quantized) concentration of q at time t when q is an intensity corresponding to the molecule q in the system under study, for h_q^t evaluated in the products and reactants region S_{pr}.[1]

The vector $h^t = [h_{q_1}^t, \ldots, h_{q_n}^t]$ with q_i an intensity of a reactant or a product, i=1,...,n summarizes the information of kinetic nature that we have about the time evolution of the (quantized) concentration of the

[1] It is important to note here that we restrict ourselves to the reactants and products region in the summation in (6). If $\mathcal{A}$ is the minimal DHA computed according to the algorithm of Section 4.3 it may happen that for some configuration C_t, $(z)C_t proj_2 = q$ for some $z \in S_{pr}$ however, we do not include those points in the summation (6).

elements q_i, i=1,...,n.

We note that in the case corresponding to the example of Figure 1,

$$h^t = [h^t_{C(1,2)}, h^t_{A(1,2)}, h^t_{B(1,2)}, <h^t_{C(1,2)}, h^t_{A(1,2)}, h^t_{B(1,2)},$$

$$h^t_{AB(1,2)}>] \qquad (8)$$

where the 3 output functions inside the < > brackets correspond to the 3 possible intensities the protein element PR(1,2) can have in the catalysis region which forms part of the reactants and products region (see Figure 3.3-12).

We note that no experimental data about the reaction systems (1,3) or (1,4) is given in Figure 1. Therefore, for this example, our identification problem consists in determining feasible time-course concentrations for the elements participating in these reaction systems.

In general, the structural identification problem consists in finding an initial activation or composite control process evolving in the DHA representing the system under study, such that the part of the output function corresponding to known kinetic data, equals the corresponding concentrations (in quantized form) at every time step. That is, let d^t_i i=1,...,n t=0,1,... be the quantized experimental concentrations of elements q_i, i=1,...,n at time steps t=0,1,..., then, we want the configurations of the process to satisfy

$$d^t_i = ((z)C_t proj_2)h^t_{q_i} = \Sigma(z)C_t proj_2 \delta^t_{q_i} \quad i=1,...,n \qquad (9)$$

$$z \in (C_t)Supp_o \cap S_{pr} \quad t=0,1,...$$

This is our fitness criterion.

Based on this criterion, we now proceed to describe the structural identification procedure; this procedure consists of 5 steps:

1 - Determination of initial time t_o and time horizon T_H of the process;

2 - Specification of a set S of feasible DHA's satisfying the fitness criterion. S is called the schemata of the identification procedure;

3 - Assignment and rearrangement of the elements of each of the DHA's of S so as to satisfy the dynamics of the known output elements;

4 - Solution of a composite control problem for each of the processes in the schemata S for which step 3 has been successfully completed;

5 - Computation of the output dynamics for the unknown elements of the process.

1 - We recall that the experimental kinetic data on which we base our identification is obtained from measurements on a cell population and not from a single cell; moreover; frequently the data spans over more than one cell generation (see Lewin [6]). Since the model developed in Section 3.3 does not consider cell division we must select an interval of the time base of the data on which to base our identification procedure. In principle, this does not imply loss of experimental data because if it spans several time generations we group it into segments with length (in time) less than one cell generation and perform an identification in each segment.

From now on, we assume that the data considered for the identification has a length within one cell generation.

The criterion for determining the initial time of the process is as follows:

a) Order the operons in the system according to their control action, i.e., if OP_i has the code for the regulator protein of OP_j the $i < j$. This ordering is carried out in the input operation of the model to the computer (see program IDHA, Section 3.5-1). The initial time of the process (for the purposes of identification) is determined by the time of appearance τ_1 of non-zero concentration for the first protein PR(1,1), and by $t_{1,1}$ its time of synthesis (see Section 3.3). The initial time for identification purposes is then determined by the formula

$$t_o = \tau_1 - 2t_{1,1} - 4 \qquad (10)$$

where all variables in (10) are evaluated in discretized units. The logic behind this convention for the initial time is that it takes $2t_{1,1} + 4$ time units to transcribe, synthesize and transport, from the synthesis to the catalysis regions, the first protein of type PR(1,1) (see Section 3.3).

For instance, in the example of Figure 1, τ_1 corresponds to the real time of 33 minutes; thus

$$\tau_1 = \left\lceil \frac{33 \times 60}{18} \right\rceil = 110 \text{ time steps.}$$

$$t_{1,1} = 10 \text{ (see Section 3.5-1)}$$

so

$$t_o = 86 \text{ time steps.}$$

Hence, our initial time step in the discretized and quantized data for this example is the 86th step. If the concentration for PR(1,1) is not part of the output function we proceed to check the zero time concentration τ_2 for PR(2,1), the first protein of the second operon OP_2, and define t_o as

$$t_o = \tau_2 - 2\, t_{2,1} - 4 - \sum_{j=1}^{n_1} t_{1,j} - 4n_1 \tag{11}$$

where n_j is the number of proteins coded by OP_1. Clearly, we can generalize this convention for the case in which the first protein for which the zero time is known is PR(j,1) for some j.

b) The time horizon T_H is determined according to

$$T_H = \tau_F - \tau_o \tag{12}$$

where τ_F is the nearest cell generation switch ($\tau_F > \tau_o$) if known or else,

where

$$\left.\begin{aligned} T_H &= \sigma - t_o \\ \sigma &\geq \sum_{i,j} 2t_{i,j} + \sum_i 4n_i \end{aligned}\right\} \tag{13}$$

where n_i = number of proteins of the i-th operon. The second criterion

(given by (13)) guarantees that all the operons of the system might be activated at least once. (Notice that in a system with no operons not all the operons need to be activated for a given process running in it.)

Finally, we select a region S of the informational space as in Section 4.5. For the example of Figure 1, we take S as determined by 5.1-20.

2 - In this step we determine a schemata S of DHA's which satisfy the given output equation.

A schemata S of DHA's is a finite set of DHA's $\{A_j\}$, each of them satisfying the fitness criterion (9). We construct every element A_j of S (using program IDHA) with the same initial time according to

a) For each time t, $t_o \leq t \leq \tau_F$, assign to the reactants and products region a sub-configuration C_t^j such that

$$((z)C_t^j \text{proj}_2)h_{q_i}^t = d_i^t \quad i=1,\ldots,n \tag{14}$$

b) Let L be the CTM of A_j (in fact, all the DHA's in S share the same CTM because this map depends only on the elements forming the system and not in their particular location in the informational space (see Section 3.5-1).

Then, we demand

$$((z)C_t^j \, L \, \text{proj}_2)h_{q_i}^{t+1} = ((z)C_{t+1}^j)\text{proj}_2 h_{q_i}^{t+1} \quad i=1,\ldots,n \tag{15}$$

That is, we demand <u>consistency</u> in the intensities corresponding to

the output function. This operation is simpler than it appears at first sight because, very few empirical rules are sufficient to satisfy (13) and once (14) is satisfied for $t = t_o, t_o + 1, \ldots, \tau_F$. In order for (15) to be satisfied, we also have a set of rules for <u>rearranging</u> C_t^j so that $C_t^j L$ is such that (15) is satisfied.

In synthesis S is a set of MDHA's A_j such that for each j, j=1,...,m

$$(C_o^j L^t)\text{proj}_2 h_{q_i}^t = d_i^t \quad t = t_o, t_o, 1, \ldots, \tau_F \quad i=1,\ldots,n$$

where C_o^j is the initial configuration assigned to A_j.

3 - In this step we perform rearranging operations on the configurations C_t^j of A_j j=1,...,m so as to satisfy (14) and (15) (the fitness criterion). We do these operations on the basis of a set of rules designed in such a way so as to make the procedure attractive from the computational point of view.

Suppose $d_i^t = \ell > 0$, the number of elements in intensity q_i that must appear in ℓ different points of the informational space in the products and reactants region. In general, if q_i is a transformation element (see Section 3.2), or, in particular, in the catalysis region. So C_t^j j=1,...,n must have, by the fitness criterion, ℓ elements in intensity q_i we insure this in the algorithm for identification, by assigning ℓ elements in intensity q_i and the corresponding structure to dormant points of the region (in the algorithm, to ℓ memory locations of the memory configuration (1 or 2 see Section 5.1, program DDHA) that holds the present con-

figuration at time t in program DDHA).

We perform the operation described in the last paragraph for every d_i^t i=1,...,n. Then we compute $C_t^j L$ (using DDHA) and test (15); if (15) is not satisfied we change the assignment for the d_i^t elements for every i, i=1,...,n for which the local condition (15) is not satisfied according to the rearrangement rules described below.

a) If q_i is a reactant in a monomolecular reaction (irreversible) (see Section 3.3) and $d_i^{t+1} = d_i^t - k \quad k < d_i^t$, this implies that in this time step, k reactions of the monomolecular type have occurred. So, be reallocating in the present memory configuration at least k q_i-elements to the catalysis region, and in the corresponding nearest neighboring locations, allocating k proteins in intensity q_i, the corresponding reactions take place at time t+1 and (15) is satisfied.

b) A similar procedure is carried out if q_i is a product in a monomolecular irreversible reaction and $d_i^{t+1} = d_i^t + k$.

c) For monomolecular reversible reactions we proceed as in a). If only the reactant appears in the output function (resp. as in b) if only the product appears in the output function). If both, reactants and product appear in the output function we must perform a) and b) simultaneously.

d) For bimolecular reactions (irreversible) (resp. reversible) we proceed similarly as in a) (resp. c) with respect to transformation elements, but in addition, if the reaction is of the form

$$q_i + q_j \xrightarrow[PR]{} q_s$$

and

$$d_i^{t+1} = d_i^t - k_i$$

$$d_j^{t+1} = d_j^t - k_j$$

we must assign k_i proteins of type PR to the catalysis region in intensity q_j, and k_j proteins of type PR in intensity k_j.

Notice that for this example the consistency condition is

$$d_s^{t+1} = d_s^t + |k_i - k_j|$$

We construct the sub-configurations $C_{t_o}^j$, $C_{t_o}^j + 1$ $\forall$ j according to the rules given above until we reach time horizon T_H, discarding along the way those A_j (schemas) for which the assignment and/or rearrangements a) - d) does not give configurations satisfying the fitness condition. Let S' be the resulting subset of S.

4 - Once at time T_H, with every one of the schemas in S' we proceed backward in time, using the procedure derived in Section 4.5 for s-lving composite process control problems for each of the schemas in S' until and if we reach an initial configuration $C_{t_o}^j$ for each of the schemas in S'. We note that each C_t^j in S' is the main stream process of the corresponding composite process control problem and the control functions (U_t^j) correspond to reactants and/or products, or proteins <u>not included</u> in the putput function.

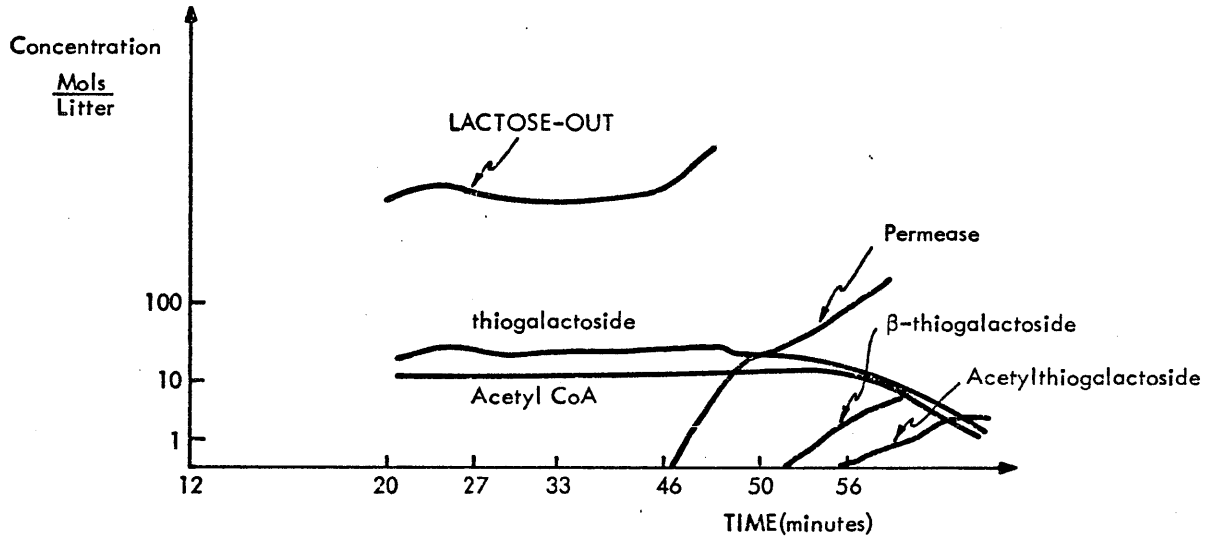
5 - Once the configurations for each of DHA's in S' has been determined, we compute the output function for those reactants and/or pro-

ducts, and proteins not included in the experimental data and average at each time step, the corresponding outputs for each of the schemas. This is the outcome of our structural identification procedure. It is clear that the larger the number of schemas considered, the more likely that we will obtain the true time-course of these concentrations simply because the concentration variables are constructed from averages of microkinetic data over a large number of samples (in the limit an ∞ number of them) and our model simulate deterministically for each case one feasible sample of the process.

In the case of our example of Figure 1 the unknown concentration time courses are the ones corresponding to the elements of reactions systems (1,3) and (1,4). We started with 10 schemas of which 6 were successful. In Figure 2 we show the computed averaged output for the six schemas with respect to real time for the chemical elements not appearing in Figure 1, so that a somparison between these two concentration time courses can be performed.

We conclude this section by remarking on an observation made in Section 3.4 about the correct level of aggregation of the model with respect to the level of aggregation of the experimental data. Notice that the data as presented to us is coarser than the dynamics of the model in the sense that it does not uniquely determine the dynamic behavior of the process representing the epigenetic process that generated this data. So, in order to reduce the cardinality of feasible processes we had to resort to our knowledge of the operation of the genetic appa-

ratus, a piece of information that can not be obtained from the data itself. So, putting together these nonobservable characteristics and the kinetic data we are able to obtain schemata that is feasible, i.e., satisfies the genetic characteristics plus the kinetic data. Notice that the schema output shown in Figure 2 satisfies with surprising precision the mass-balance equation for each of the two nonobserved reaction systems.



A Schema of Non-observable Reaction Time Course for the Lac-Operon in the Structural Identification Procedure

Figure 2

6. SUMMARY, CONCLUSIONS AND FUTURE RESEARCH

6.1 Summary

In this thesis, we have developed an operational model of the epigenetic apparatus of procaryotes on the basis of the fundamental dogma of molecular genetics and the Jacob and Monod operational model of operon-systems. The model, a computational device termed a Distributed Hierarchical Automaton (DHA), posseses the following general characteristics:

1 - It is a composite of elements represented by discrete-time, finite, state output automata allocated in a descrete Lattice (Z^K) called the informational space.

2 - The dynamics of the model are completely determined by the local state transition of its elements (see Section 4.1).

3 - The state transition of an element in the model is functionally dependent on the state of its nearest neighboring elements in the informational space (see Sections 3.1 and 4.1).

4 - At any time the state of all but a finite number of elements is the dormant state whose operational significance is "no-information is carried by the element at present time".

5 - The state of every element is composed of two hierarchies of date: structure and intensity. The latter represents the energetic status of the element at the operational level of aggregation (see Sections 3.1 and 3.4) and represents the only part of the state that is

observable from kinetic-based experimental observation (see Section 5.2). The former represents those molecular characteristics of the element which determine its functionality in the genetic apparatus.

6 - The model is derived in six steps:

a) Classification of the molecules or actions of the genetic apparatus into a finite set of classes (see Section 3.2).

b) Determination of a <u>type</u>[1] for each of the classes in step a), on the basis of the generic properties of the elements of each class (see Section 3.3).

c) Allocation of the elements to specific <u>regions</u> of the informational space $I = Z^3$ (see Section 3.3).

d) Determination of a universal type capable of simulating the state dynamics of any of the elements of the model (see Section 4.4) obtained in step b).

e) Simplification of the model obtained in step d) by the construction of a minimal-state DHA whose time-computational complexity is the lowest possible (see Section 4.3).

f) Algorithmic implementation of the model in a digital computer (see Sections 3.5.1 and 5.1).

Two applications of the model in the study of epigenetic processes

[1] i.e., a characteristic element. By determination of a type, we mean, determination of its state transition structure.

were considered:

a - Analysis of three types of canonical epigenetic control processes: Initial activation processes, composite processes and construction processes (see Section 4.5).

b - Structural identification of unknown kinetic characteristics of epigenetic control processes on the basis of experimental kinetic data about these processes (see Section 5.2), i.e., given the concentration time courses of some of the chemical elements in the process, find the concentration time course for the rest of them.

We conclude this summary with an observation about the generality of the model.

The model has been derived with the fundamental criterion of serving as a tool for studying a variety of epigenetic control processes rather than tailored to a specific process. With this criterion in mind, we developed a construction program (IDHA in Section 3.5) so that the user can specify the specific characteristics of the process under study and the procedure constructs a representation of the model in an effective, sequential manner. This allows the use of the model in epigenetic studies by researchers in molecular genetics who do not have a computer science or automata theory background.

6.2 Conclusions

We have found that the distributed hierarchical automaton possesses dynamic characteristics which make it a very useful tool in the simula-

tion and analysis of complex epigenetic processes in procaryotes.

The DHA is well suited for digital computer implementation and is rich in structural properties so that a great variety of epigenetic processes can be implemented with it.

With respect to other computer based models of epigenetic control processes such as Dendral [1] or Ziegler's model of E.coli, [2], our model has the unique advantage of representing functional characteristics of the elements and their (local) dynamic interaction. Also, the state dynamics of the elements are derived on the basis of a detailed analysis of the operational characteristics of these elements in the cell and not on a formal analogy with makrokinetic models of chemical reactions or based on heuristic considerations as some of the models mentioned above are.

As a consequence of the property of our model mentioned above, our model has the unique characteristic of being a discrete finite state model, and it captures the fundamental characteristics of the processes it simulates.

We have also found, that the DHA admits an algorithmic realization based on a generalization of the corresponding algebraic procedure for ordinary automata and, the realization procedure and its associate procedures developed in Chapter 4 represent, to our knowledge, the first instance in which such types of (algebraic) construction has been applied to distributed automata.

In synthesis, the importance of the model developed in here resides

in its use as a tool for predicting the structure of the kinetics of (multi) operon systems, the identification of the characterisitcs of the kinetics of nonmeasured parts of these kinetics on the basis of (partial) measurements of concentration time courses of some of the elements in the system and finally, the models provide an efficient tool in which to test hypothesis about the characteristics of epigenetic processes such as transcriptional control of some or all operons in the system, average time before the destruction of MRNA's, ribosomes, etc.

6.3 Critique and Future Research

In this section we give an overall evaluation of the results presented in this thesis and suggest several directions of future work in the further development of these results.

Although the formal model, developed in the thesis is capable of handling fairly complex operon systems, in the simulation of their structural characteristics in time, we believed it to be only a first step in the development of a comprehensive computational model for the analysis and reliable behavioral prediction of epigenetic control processes in procaryotes.

Specifically, we believe an important improvement in the model can be attained in the representation of genes and protein elements (see Section 3.3) in the following direction. Recall that we represented the information content of a gene by a single number (the time of synthesis) rather than a representation of the sequence of codons confroming it.

In order to obtain a model whose dynamics resembles as closely as possible the transcription part of epigenetic processes the types representing operons must be modified so that their corresponding genes do simulate the language characteristics of the genetic code and its transcription--i.e., we may want types for individual codons. A similar observation is valid with respect to the representation of systhesis of proteins, i.e., the representation of the 20 different amino-acid residues that form a protein and its incorporation during synthesis instead of our present representation in which we only count the number of amino acids in a protein and not the sequence of amino acids. These modifications will not alter the state structure of our model, but will increase its computational complexity. Nevertheless, we believe that the improvement in the closeness with which the model dynamics resembles the dynamics of the real epigenetic process is worth this disadvantage.

We believe that a more complex operon system than the one utilized for illustrative purposes in the thesis (the lac operon) should be implemented with the model in order to determine its capabilities and practical constraints (i.e., computational time, simulation errors, etc.). We mention that although only the example of the lac operon is reported here we have tested the model with other, single-operon systems with encouraging results with respect to the ability of the model to predict kinetic behavior of the molecules participating in the processes resulting from the activation of these operons. (A particularly interesting result was obtained in the structural identification of the arabinose

operon.)

Finally, with respect to the structural identification, we think, that a definite proof of the capabilities of the model can be assessed by running the procedure for a system for which the kinetics is known completely from experimental measurements. The acid test of this model and its extension will be its ability to predict accurately experimentally observed data. This clearly is the most important next step.

BIBLIOGRAPHY

CHAPTER 1.

1. Rosen, R., Mechanics of Epigenetic Control in "Principles of Mathematical Biology," R. Rosen, Ed., 1, Academic Press, 1973.

2. Wolpert, L., "Positional Information and the Spatial Pattern of Cellular Differentiation," J. Theor. Biol., 25, pp. 1-47, 1969.

3. Watson, J. D., Molecular Biology of the Gene, Benjamin Press, 1970, 2nd Ed.

4. Beckwith, J. R. and D. Zipser (Ed.), The Lactose Operon, Cold Spring Harbor Laboratory, Cold Spring Harbor, 1970.

5. Jacob, F. and J. Monod, "Genetic Regulatory Models in Synthesis of Proteins," J. of Mol. Biol., 3, p. 318, 1961.

6. Lewin, B., Gene Expression - 1: Bacterial Genomes, John Wiley, 1974.

7. Rosen, R., Hierarchical Organization in Automata--Theoretical Models of Biological Systems, in "Hierarchical Structures", L. L. Whyte, A. G. Wilson and D. Wilson, eds., American Elsevier, 1969.

8. J. D. Watson, Biology 108 Course Notes, Harvard University, 1971.

9. Ptäshne, E. and E. Riordan, "Initiation and Termination Control of Genetic Actions," American Scientific, 200, 1976.

10. Gallager R. and E. Allen, Computer Architecture Course Notes (6.332) M.I.T., 1974.

11. Prigogine, I., Thermodynamics of Irreversible Processes, John Wiley, 1967, 3rd Ed.

12. Ziegler B. P., "Introduction to a Theory on Modeling and Simulation," Proceedings Sixth Annual Pittsburgh Conference on Modeling and Simulation, University of Pittsburgh, 1975.

13. Eilenberg, S., Automata Languages and Machines, Vol. A, Academic Press, 1973.

14. Reiner, J. M., The Organism as an Adaptive Control System, Prentice Hall, 1968.

15. Codd, R., Cellular Automata, ACM Publication, Academic Press, 1968.

CHAPTER 2.

1. op cit [1] Bibliography of Chapter 1.

2. op cit [2] Bibliography of Chapter 1.

3. Pattee, H. H., The Nature of Hierarchical Control in Living Matter, in "Foundations of Mathematical Biology, Vol. 1, Subcellular Systems," Academic Press, 1973.

4. Bartholomay, A. F., Chemical Kinetics and Enzyme Kinetics in "Foundations of Mathematical Biology, Vol. 1, Subcellular Systems," Academic Press, 1973.

5. Gutfreund, H., Enzymes: Physical Principles, J. Wiley, 1972.

6. Wood, W. B., Wilson, J. H., Benbow, R. M. and L. Hood, Biochemistry Problems Approach, Benjamin Press, 1974.

7. Lehninger, A. L., Biochemistry, Worth, 1970.

8. Banks, L.,"Kinetics of Metabolic Cascades," J. of Theor. Biol., 18, p. 261, 1975.

9. Davidson, R.,"The Control of Metabolites in E. Coli", Proceedings IFAC, 1974.

10. Davidson, R., Personal Communication (1976).

11. Goodwin, I., Thalandic Mechanisms in Genetic Systems, in Towards a Theoretical Biology, H. Waddington (Ed.), Vol. 1, 1968.

12. op cit [1] Bibliography of Chapter 1.

13. Hood, L. E., Wilson, J. H., Wood, W., Molecular Biology of Eucaryotic Cells, A Problems Approach, Vol. 1, Benjamin Press, 1974.

14. Goodwin, E., and Ziegler, B. P.,"Adaptive Behavior of Simulated Bacterial Cells Subject to Nutritional Shifts," J. of Theor. Biol., 29 (1), 1970.

15. Bartholomay, A. F., Chemical Kinetics and Enzyme Kinetics in Foundations of Mathematical Biology, R. Rosen (Ed.), Vol. 1, Academic Press, 1973.

16. Higgins, J. J., "The Theory of Oscillating Reactions," Ind. Eng. Chem. 59, pp. 18-62, 1967.

17. Slater, N. B., Theory of Unimolecular Reactions, Vol. 1, Cornell University Press, 1959.

18. Slater, N. B., Theory of Unimolecular Reactions, Vol. 2, Cornell University Press, 1959.

19. op cit [6] Bibliography of Chapter 1.

20. op cit [3] Bibliography of Chapter 1.

21. White, A., Handler, P. and E. L. Smith, Principles of Biochemistry, McGraw-Hill, (5th Ed.), 1973.

22. Holland, J. H., Adaptation in Natural and Artificial Systems, The University of Michigan Press, 1975.

23. Waddington, C. H., Towards a Theoretical Biology, Introduction, Vol. 1, Edinburgh University Press, 1968.

24. Goodwin, B. C., Temporal Organization in Cells: A Dynamic Theory of Cellular Control Processes, Academic Press, 1963.

25. Kauffman, A. C., Notes on Mathematical Biology #5, John Wiley, 1971.

26. op cit [14] Bibliography of Chapter 1.

26a. op cit [23] Bibliography of Chpater 3.

27. Arbib, M. A., Self Reproducing Automata-Some Implications for Theoretical Biology, in Towards a Theoretical Biology, 2 sketches, Edinburgh University Press, 1969.

28. op cit [23] Bibliography of Chapter 2.

29. Lewontin, M. R., "Some Theoretical Aspects of Genetic Activity in Procaryotes," J. Theor. Biol., 12, pp. 136-158, 1972.

30. Burns, R., "Computer Simulations of Genetic Dynamics in Bacteria," J. Theor. Biol., 14, pp. 227, 251, 1973.

31. op cit [23] Bibliography of Chapter 2.

32. Engeler, D., Principles of Computation Theory, Addison-Wesley, 1971.

33. Watson, A. M., "Formal Languages in Genetic Circuits," J. Theor. Biol., 2, pp. 117-126, 1971.

34. Levin, D. H., "Local Interaction Between Spatially Distributed Biological Entities, J. Theor. Biol., 3, pp. 185-197, 1972.

35. Monod, J. and F. Jacob, "General Conclusions: Teleonomic Mechanism in Cellular Metabolism, Growth and Differentiation," Cold Spring Harbor Symposium, Quant. Biol. Cellular Regulatory Mechanisms, 26 pp. 384-401, 1965.

36. Lewin, B., The Molecular Basis of Gene Expression, John Wiley, 1970.

37. Pattee, H., The Challenge of Complex Systems in Hierarchy Theory, Braziller Press, 1973.

38. op cit [37]

39. Ziegler, B. P., Theory of Modeling and Simulation, John Wiley, 1976.

40. Stone, H., Computer Architecture of Parallel Processes, in Computer Architecture, H. Stone, Ed., John Wiley, 1976.

41. Tannenbaum, L., Structured Computer Organization, John Wiley, 1975.

42. Crick, F. H. C., "Central Dogma of Molecular Biology," Nature, 227, pp. 561-562, 1970.

43. Watson, J. D. and F.H.C. Crick, "Genetic Implications of the Structure of DNA," Nature, 171, pp. 964-967, 1953.

44. Malkin, L. I. and A. Rich, "Partial Resistance of Nascent Polypeptide Chains to Proteolytic Digestion Due to Ribosome Shielding," J. Mol. Biol, 26, pp. 329-346, 1967.

45. Bartholomay, A. F., "Molecular Set Theory: A Mathematical Representation for Chemical Reaction Mechanisms," Bull. Math. Biophys., 22, p. 285, 1960.

46. Prigogine, I. and R. Brout, "Irreversible Processes in Weakly Coupled Systems," Proceedings of the International Symposium on Transport Theory in Statistical Mechanics, Brussels, 1956.

47. Eigen, M., "Self Organization of Matter and the Evolution of Biological Macromolecules, Naturwissenschaften, 58, pp. 465-523, 1971.

48. Jacob, F. and J. Monod, "Genetic Regulatory Mechanisms in the Synthesis of Proteins," J. Mol. Biol., 3, pp. 318-356, 1961.

49. Davis, B. D., "Role of Subunits in the Ribosome Cycle," Nature, 231, pp. 153-157, 1971.

50. Wood, W. B., Wilson, H. J., Benbow, R. M. and L. E. Hood, Biochemistry: A Problems Approach, Benjamin Press, 1974.

51. op cit [3] Bibliography of Chapter 3.

52. Oppenheim, A. and F. Schaffer, Digital Signal Processing, John Wiley, 1975.

53. Turing, A. M., "The Chemical Basis of Morphogenesis," Phil. Trans. Roy. Soc., B237, pp. 5-72, 1952.

54. Hayes, H.,"Purification of E. Coli Leucine Suppression," J. Bio. Chem., 246, pp. 4951-4954, 1971.

CHAPTER 3.

1. Aoki, M., "State Aggregation of Dynamical Systems," A. C. Trans., 18, pp. 1057-1072, 1971.

2. Hanna, M. W., Quantum Mechanics in Chemistry, 2nd Ed., Benjamin Press, 1969.

3. Calvert, J. G. and J. N. Pitts, Photochemistry, John Wiley, 1966.

4. Turro, N., Photochemical Reactions, Benjamin Press, 1968.

5. Dickerson, R. E., Molecular Thermodynamics, Benjamin Press, 1969.

6. Ginsburg, I., Automata Theory, Academic Press, 1968.

7. Buchanan, R. and R. L. Isaacson, "Dendral a Program for Genetic Inference," Machine Intelligence, 6, pp. 118-142, 1970.

8. Roitt, M., "Energetic Structure of Large Macromolecules," J. of Spect., 6 pp. 212-228, 1972.

9. op cit [3] Bibliography of Chapter 1.

10. Baltimore, J. and A. Spector, "The Mechanisms of Inverse Transcriptase," J. Theor. Biol., 6 pp. 112-115, 1968.

11. op cit [6] Bibliography of Chapter 1.

12. Sampson, B. L., Introduction to the Theory of Adaptation, McGraw-Hill, 1972.

13. op cit [1] Bibliography of Chapter 1.

14. op cit [47] Bibliography of Chapter 1.

15. op cit [14] Bibliography of Chapter 1.

16. op cit [39] Bibliography of Chapter 2.

17. op cit [5] Bibliography of Chapter 2.

18. Byam, L., Photochemistry, Benjamin Press, 1971.

19. op cit [42] Bibliography of Chapter 2.

20. Katchalsky, K., "The Irreversible Thermodynamics and Information Theory in Non-Stationary Processes," J. Molè. Bio., 213, pp. 525-545, 1964.

21. op cit [31] Bibliography of Chapter 1.

22. op cit [1] Bibliography of Chapter 4.

CHAPTER 4.

1. Bobrow, L. S. and M. A. Arbib, Discrete Mathematics: Applied Algebra for Computer and Information Science, W. B. Saunders, 1974.

2. Eilenberg, S., Automata Languages and Machines, Vol. A, Academic Press, 1974.

3. Ginsburg, A., Algebraic Theory of Automata, Academic Press, 1968.

4. Arbib, M. A., Abstract Automata, Academic Press, 1968.

5. Aho, A. V. and J. D. Ullman, The Theory of Parsing Translation and Compiling, Vol. 1, Prentice Hall, 1973.

6. Conway, J. H., Regular Algebra and Finite Mechanics, Chapman and Hall, 1971.

7. Cohn, P. M., Universal Algebra, Harper and Row, 1965.

8. Lang, S., Algebra, Addison Wesley, 1971.

9. Lausch, H. and Nobauer, W., Algebra of Polynomials, North-Holland/ American Elsevier, 1973.

10. Foster, C. C., Content Addressable Parallel Processors, Van Nostrand Reinhold, 1976.

11. Miller, R. E., Switching Theory, Vol. I, Combinatorial Circuits, John Wiley, 1966.

12. Abian, A., "On the Solvability of Infinite Systems of Boolean Polynomial Equations," Colloq. Math., 21, pp. 27-30, 1970.

13. Rudeanu, S., An. Sti. Univ. IQSI Sect. I a Mat. 6, 1960.

14. Goodwin, S. A., STARAN S APPLE; AN Hybrid Associative Computer, Rome Air Development Center RADC-TDR-65, 1965.

15. op cit [2] Bibliography of Chapter 1.

16. op cit [1] Bibliography of Appendix A.1

17. Birkhoff, G., "Universal Algebra and Number Theory," SIAM, 12, pp. 120-165, 1975.

18. Aho, A. V., et al, Algorothms, McGraw-Hill, 1975.

19. Stone, H., Discrete Mathematics, AMF Publ., 1974.

20. Knuth, A., Semi-Numerical Algorithms, Vol. 1, J. Wiley, 1970.

21. Treves, F., Topological Vector Spaces, Distributions and Kernels, Academic Press, 1967.

22. Birkhoff, G. and Bartee, L., Applied Modern Algebra, Academic Press, 1975.

23. Boole, L., "Symbolic Mathematics," J. Applied Math., (Reprint) p. 1898, 1854.

24. Hadley, R., Dynamic Programming, J. Wiley, 1964.

CHAPTER 5.

1. op cit [5] Bibliography of Chapter 1.

2. op cit [4] Bibliography of Chapter 2.

3. op cit [5] Bibliography of Chapter 3.

4. Kohn, W. and Johnson, T. L., "Interfacial Electrochemistry and Electrodynamics of Ionic Media," ESL-P-697, M.I.T., 1977.

5. op cit [6] Bibliography of Chapter 2.

6. op cit [6] Bibliography of Chapter 1.

CHAPTER 6.

1. Dendral, J., Physical Structure in System Theory, Academic Press, 1974.

2. op cit [12] Bibliography of Chapter 1.

APPENDIX A.1.

1. Gratzer, G., Universal Algebra, Van Nostrand/Renihold, 1968.

2. Cohn, M. P., Universal Algebra, Harper and Row, 1965.

3. Kuros, A. G., The Theory of Groups, Vol. II, 2nd English Ed., Chelsea Publishing Co., 1960.

4. Hule, H., On Polynomials and Algebraic Functions in Universal Algebras, Ph.D. Thesis, U. of Vienna, 1968.

APPENDIX A.1

UNIVERSAL ALGEBRAS

In this appendix, we give a short review of those aspects of universal algebras that are used in the formulation and study of DHA's in Chapter IV and Section 3.4. A more complete description and analysis of universal algebras may be found in Gratzer [1], Cohn [2], Kuros [2], and Hule [4].

A universal algebra Q is a pair $\langle Q, \Phi \rangle$[1] where Q is a nonempty set and $\Phi = \{f^{\gamma} | \gamma \in G\}$ is a family of operations on Q being indexed by the set of ordinals $i<\theta$ where θ is an arbitrary ordinal.

The family $\tau = \{n_{\gamma} \geq 0 | f^{\gamma}$ is an n_{γ}-ary operation, $\gamma \in G\}$ is called the type of Φ.

Two algebras Q, Q' are said to be similar (denoted by Q s Q') if they are of the same type.

Example 1: Let $\Phi = \{f^1, f^2, f^3\}$, $\tau = \{2, 1, 0\}$. Let Q be a nonempty set. Define f^i i=1,2,3 as follows: $\forall\, q_1, q_2, q_3 \in Q$

$$(q_1,q_2)f^1 = (q_2,q_1)f^1$$

$$((q_1f^2,q_1)f^1,q_3)f^1 = (q_1,(q_2,q_3)f^1)f^1$$

$$(q_1f^2,q_1)f^1 = (q_1,q_1f^2)f^1 = f^3$$

---

[1]We use the same symbol to denote the algebra and its set. In each instance we will make clear which of them is meant.

$$\exists\, q_o \in Q,\ f^3 \equiv q_o$$

$$(q_1, f^3) f^1 = q_1 = (f^3, q_1) f^1$$

with these definitions $\langle Q, \Phi \rangle$ is an abelian group with f^1= addition, f^2= negation, f^3= identity element. Clearly all abelian groups are similar (in fact all groups are similar). ▲

Notation: For similar algebras Q, Q' we used the same operation symbol ($f^\gamma \in \Phi$) for those operations on Q and Q' which are indexed by the same $\gamma \in G$. ▲

If Q is an algebra, the cardinal $|Q|$ of its set Q, is called the order of the algebra Q.

Let $\langle Q, \Phi \rangle$ be a universal algebra; let $U \neq \emptyset$, $U \subseteq Q$ such that $\forall\, \gamma \in G$

$$(q_1, \ldots, q_{n_\gamma}) f^\gamma \in U$$

whenever $(q_1, \ldots, q_{n_\gamma}) \in U$ for $n_\gamma > 0$, and $f^\gamma = \hat{q} \in U$ if $n_\gamma = 0$ then, the restriction of any $f^\gamma \in \Phi$ to U yields an operation on U which we denote again by f^γ. The algebra $\langle U, \Phi \rangle$ is called a subalgebra of Q and Q an extension of U.

As a consequence of the definition above we have:

Proposition 1: Any nonempty intersection of subalgebras of a given algebra Q is again a subalgebra of Q. ▲

If S is a nonempty subset of Q, $\langle Q,\Phi\rangle$ an algebra, then, the intersection of all subalgebras of $\langle Q,\Phi\rangle$ containing S is called the subalgebra of $\langle Q,\Phi\rangle$ generated by S and denoted by $[S]\Phi$ or simply [S].

Let Q, Q' be similar algebras. A homorphism from Q to Q' is a mapping $v:Q \to Q'$ such that

$$(a_1,\ldots,a_{n_\gamma})f\,v = (a_1v,\ldots,a_{n_\gamma}v)f^\gamma$$

$\forall\ \gamma \in G$. Such that $n_\gamma > 0$ and

$$f^\gamma v = f^\gamma$$

$\forall\ \gamma$ such that $n_\gamma = 0$.

If v is injective, it is called a monomorphism; if its is surjective it is called an epimorphism; if it is bijective it is called an isomorphism.

The following lemma establishes the basic characteristics of algebra homomorphisms.

Lemma 1:

(i) If v is a homomorphism from $\langle Q,\Phi\rangle$ to $\langle Q',\Phi\rangle$, then $\langle Qv,\Phi\rangle$ is a subalgebra of Q'.

(ii) If v is an isomorphism from $\langle Q,\Phi\rangle$ to $\langle Q',\Phi\rangle$ and η is an isomorphism from $\langle Q',\Phi\rangle$ to $\langle Q'',\Phi\rangle$ then so is $v\eta$ from $\langle Q,\Phi\rangle$ to $\langle Q'',\Phi\rangle$.

(iii) If v is an algebra isomorphism then so is v^{-1}.

(iv) The relation "Is isomorphic to" in any class of similar algebras, is an equivalence relation. ◀

Corollary. Let $\langle Q,\Phi\rangle$ be a universal algebra of finite order (i.e., $|Q|<\infty$). Then any monomorphism and any epimorphism from Q to Q, is an isomorphism. (Sometimes authors call this an automorphism.) ▲

Let $\langle Q,\Phi\rangle$ be an algebra. An equivalence relation Θ on Q is called a congruence on Q if for any $f^\gamma \in \Phi$ with $n_\gamma > 0$, $q_1 \Theta q_1', \ldots, q_{n_\gamma} \Theta q_{n_\gamma}'$ ($q_i, q_i' \in Q$ $i=1,\ldots,n_\gamma$), implies

$$(q_1,\ldots,q_{n_\gamma})f^\gamma \Theta (q_1',\ldots,q_{n_\gamma}')f^\gamma$$

Let $\tilde{Q}$ be the set of equivalence classes under Θ, and let $(q)E$, $q \in Q$, denote the class in $\tilde{Q}$ containing q. On $\tilde{Q}$ we define for every $\gamma \in G$ an n_γ-ary operation f^γ as follows: For $q_1,\ldots,q_{n_\gamma} \in Q$

$$\begin{aligned} (q_1)E,\ldots,(q_{n_\gamma})E\ f^\gamma &= (q_1,\ldots,q_{n_\gamma}f^\gamma)E \quad \text{for } n_\gamma > 0 \\ f^\gamma &= (f^\gamma)E \qquad\qquad \text{for } n_\gamma = 0 \end{aligned} \tag{1}$$

The operations f^γ, $\gamma \in G$ on Q are well defined since the result of any operation does not depend on the particular choice of representatives $\langle\tilde{Q},\Phi\rangle$ is called the factor algebra of Q with respect to Θ and is denoted by $Q|\Theta$.

From (1) we deduce that the mapping $v:Q \to Q|\Theta$ defined by

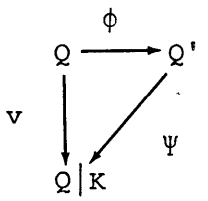
$$qv = (q)E$$

is an epimorphism. It is called the canonical epimorphism from Q to $Q|\Theta$. Thus, $Q|\Theta$ is a homomorphic image of Q.

Theorem 1: (Homomorphism theorem, Gratzer)

Let Q, Q' be similar algebras. Let $\phi: Q \to Q'$ be an epimorphism of algebras. Then there exists a congruence K on Q and an isomorphism $\Psi: Q' \to Q|K$ such that $\phi\Psi = v$ is the cononical epimorphism from q to Q K.

The statement of the theorem is illustrated with the following cummutative diagram:



Outline of the proof:

We define K by;

$$q_1 K q_2 \iff q_1\psi = q_2\psi$$

It is easy to show that K is a congruence; moreover the mapping $q' \to q'\phi^{-1}$ is clearly a bijection from Q' to $Q|K$ and it is also clear that the diagram commutes. We show next, that this mapping, denoted by ψ is a homomorphism.

Let $q'_j = q_j\psi$, $j=1,\ldots,n\gamma$, $n\gamma > 0$, then since ψ and v are homomorphisms

$$(q'_1,\ldots,q'_{n_\gamma})f^\gamma\psi = (q_1 v,\ldots,q_{n_\gamma} v)f^\gamma$$

and it only remains to observe that commutativity of the diagram implies

that

$$q_i'\Psi = q_i v \quad i=1,\ldots,n_\gamma$$

If $n_\gamma = 0$, then since ψ and v are homomorphisms we must have

$$f^\gamma\Psi = f^\gamma$$ ▲

Remark: If ϕ as defined in Theorem 1, is an algebra homomorphism, then by changing the range of ϕ from Q' to $Q\phi$ we obtain an epimorphism from Q to $Q\phi$ which we also denote by ϕ. The set $\{q'\phi^{-1} | q' \in Q\phi\}$ induces a congruence K on Q. K is called the Kernel of ϕ and usually it is denoted by Ker ϕ. ▲

Every algebra Q has at least two congruences: the congruence where the classes consist of one element and the congruence whose only class is Q itself. In case there are no more congruences on Q we say that Q is a simple algebra.

In view of Theorem 1 and the definition of simple algebras above we have;

Proposition 2: $\langle Q,\Phi\rangle$ is simple if every homomorphism of Q is a monomorphism or maps to the algebra $\langle\{q\},\Phi\rangle$ (the algebra of order 1). ▲

Let $L(Q)$ be the let of all congruences in the algebra Q. On $L(Q)$ we define a binary relation $\leq$ as follows

$$\Theta_1 \leq \Theta_2 \iff \Theta_1 \subseteq \Theta_2 \qquad (2)$$

with Θ_1, Θ_2 considered as subsets of the cartesian product $Q \times Q$.

Theorem 2: The set $L(Q)$ is a complete lattice with respect to $\leq$ (this lattice is called by Birkoff [ ], the congruence lattice of Q).

Proof:

See Gratzer [ ]. ▲

Given a set $M = \{\Theta_i, i \in I\}$ in $L(Q)$, the greatest lower bound of M is the congruence Δ on Q defined by

$$\Delta = \bigcap_{i \in I} \Theta_i \tag{3}$$

i.e., $\forall\, q_1, q_2 \in Q$, $q_1 \Delta q_2 \iff q_1 \Theta_i q_2 \quad \forall\, i \in I$.

The least upper bound of M, is the congruence Ω on Q defined as follows:

$$q\ \Omega\ q' \iff \exists\, \Theta_{i_1}, \ldots, \Theta_{i_r} \in M \text{ and } q_1, \ldots, q_{r-1} \text{ in } Q \text{ such that}$$

$$q\ \Theta_{i_1} q_1,\ q_1\ \Theta_{i_2} q_2, \ldots, q_{r-1}\ \Theta_{i_r} q' \tag{4}$$

i.e., the smallest congruence containing $\bigcup_{i \in I} \Theta_i$. We call (4) a chain of length r from q to q'.

We now want to give some results with respect to the relations between congruences on an algebra Q and those on its factor algebra $Q|\Theta$ ($\Theta \in L(Q)$). These are given by:

Theorem 3: (Second Isomorphism theorem, Birkoff [ ]).

Let Q be an algebra Θ a congruence on Q, $(q)E$ the congruence class containing $q \in Q$ and $\mathcal{D}$ the sublattice of $L(Q)$ consisting of all congruences Ω, $\Theta \leq \Omega$ on Q then:

(i) If $\Omega \in \mathcal{D}$, then there exists a congruence $\Omega|\Theta \in L(Q|\Theta)$ defined by

$$(q)E\ \Omega|\Theta(q')E \text{ iff } q\ \Omega\ q'$$

(ii) If $\Sigma \in L(Q|\Theta)$ then there exists $\overline{\Sigma} \in \mathcal{D}$ defined by

$$q\ \overline{\Sigma}\ q' \text{ iff } (q)E\ \Sigma(q')E.$$

(iii) The mapping $\alpha:\mathcal{D} \to L(Q|\Theta)$ defined by

$\Omega\alpha = \Omega|\Theta$ is a lattice isomorphism and $\Sigma\alpha^{-1} = \overline{\Sigma}$ (see (ii))

(iv) $Q|\Omega$ is isomorphic to $(Q|\Theta)\,|\,(\Omega|\Theta)$ for any $\Omega \in \mathcal{D}$.

Some of the results of theorem 3 are used in Section 3.4.

Now we consider two operations on classes of similar algebras; direct product and direct limit.

Let $\{\langle Q_j, \Omega\rangle,\ j \in I\}$ I an ordered set, be a family of similar algebras. Let $Q = \underset{j \in I}{X}\, Q_j$ be the cartesian product of the sets $\{Q_j\}, j \in I$. We denote any element q of Q as a function $I \to Q$ with $(j)q = q_j \in Q_j$. We define the operations $f^{\gamma} \in \Omega$ on Q as follows:

$$\left.\begin{aligned} &(\{(j)q_1\}\ldots\{(j)q_{n\gamma}\})f^\gamma = \{(j)q_1,\ldots,(j)q_{n\gamma}f^\gamma\} \text{ if } n\gamma > 0 \\ &\text{and} \\ &f^\gamma = \{(j)f^\gamma\} \quad \text{if } n\gamma = 0 \end{aligned}\right\} \quad (5)$$

where $\{(j)q_i\}$ denotes

$$((j_1)q_i,\ldots,(j_n)q_i\ldots) \quad \forall\, j_k \in I$$

The algebra $\langle Q,\Omega\rangle$ is called the direct product of the algebras Q_j and it is denoted by $\prod_{j\in I} Q_j$. We will denote the operation set of Q by Ω and its operations by f^γ, $\gamma \in G$.

We note that the definition of $\prod_{j\in I} Q_j$ given above implies that for fixed $j \in I$ the mapping

$$e_j^I : \{(i)a\} \to (j)a \in Q_j \text{ is an epimorphism from } \prod_{i\in I} Q_j \text{ to } Q_j.$$

The mappings e_j^I, $j \in I$ are called the projection mappings.

Direct products of algebras are used in Section 4 in the construction of covering DHA's.

Let I be the set of all ordinals $j < \varepsilon$ where ε is an arbitrary ordinal. A family of similar algebras $\{\langle Q_j,\Phi\rangle,\ j \in I\}$ is called an ascending family of algebras if Q_k is a subalgebra of Q_ℓ whenever $k < \ell$. We set $Q = \bigcup_{j\in I} Q_j$ and define the operations $f^\gamma \in \Phi$ on Q as follows:

$$f^\gamma = f^\gamma \text{ of } Q_o \quad \text{if } n_\gamma = 0$$

and,

$$\left.\begin{array}{l} \text{for } n_\gamma > 0, \text{ let } \mu = \min j \in I \\ \qquad \text{subject to } q_1,\ldots,q_{n\gamma} \in G_\mu \end{array}\right\} \tag{6}$$

Then,

$$q_1,\ldots,q_{n\gamma}f^\gamma = q_1,\ldots,q_{n\gamma}f^\gamma \text{ in } \langle G_\mu,\Phi\rangle \tag{7}$$

The algebra $\langle Q,\Phi\rangle$ is called the direct limit(or direct product, second form in Gratzer) of the ascending family $\{\langle Q_j,\Phi\rangle, j \in I\}$.

Lemma 1: Let $\{\langle Q_j,\Phi\rangle, j \in I\}$ be an ascending family of algebras. Let $\langle Q,\Phi\rangle$ be its direct limit, then Q_j is a subalgebra of Q for every $j \in I$. ▲

APPENDIX A.2

In this appendix, a listing of the program for computing the trim part of DHA A is provided. The description of its operation is given in Sub-section 4.2.1.

```
PROGRAM ACCESSIBLE
**********************************************
THIS PROGRAM COMPUTES THE ACCESSIBLE PART OF A
GIVEN M-MODULE FROM A GIVEN INITIAL PROGRAM
STEP SET IG AND INITIAL STATE SET IQ.
**********************************************
 FRACTION A(2500),B(2500)
**********************************************
A IS THE ARRAY OF ACCESSIBLE PAIRS IN EACH TIME
TRANSITION B IS THE SET OF ACCESSIBLE PAIRS
FROM IG,IQ
**********************************************
INITIALIZE A
**********************************************
DO 10 I=1,2500
B(I)=-1
A(I)=-1
**********************************************
INIT READS-IN INITIAL PAIRS AS A ONE DIMENSIONAL
ENCODING.
**********************************************
CALL INIT(IG,IQ,N,M,A)
DO 15 I=1,N*M
B(I)=A(I)
**********************************************
N=CARDINALITY(IG) M=CARDINALITY(IQ)
**********************************************
N3=N*M
DO 20 I=1,N3
IF(A(I))20,30,30
CONTINUE
GO TO 90
**********************************************
UPDATE A : COMPUTE THE NEW PAIRS OF ACCESSIBLE
PROGRAM STEPS AND STATES FROM THE SET OF PAIRS
A.
**********************************************
CALL NEWA(A,A,N1,M1)
```

```
************************************
CONTINUATION OF PROGRAM ACCESSIBLE
************************************************
UPDATE B THE SET OF ACCESSIBLE PAIRS
N2=NUMBER OF NEW ACCESSIBLE PAIRS
Nl=NUMBER OF NEW ACCESSIBLE PROGRAM
STEPS
N2=NUMBER OF NEW ACCESSIBLE STATES
************************************************
I=0
N2=Nl*Ml
IF(N2.EQ.0) GO TO 90
IF(I.EQ.N2) GO TO 120
I=I+l
DO 50 J-l,N3
IF(A(I).EQ.B(J)) GO TO 75
CONTINUE
GO TO 98
DO 100 K=I,N2-l
A(K)-A(K+l)
A(N2)=-l
N2-N2-l
GO TO 98
K=l
DO 150 I=N3+l,N3+N2+l
B(I)=A(K)
K=K+l
N3=N3+N2+l
GO TO 30
************************************************
FACTOR COMPUTES THE ACCESSIBLE PROGRAM STEPS G
AND THE ACCESSIBLE STATES Q
************************************************
CALL FACTOR(B,N3,G,Q)
CALL EXIT
END
```

APPENDIX A.3

In this appendix we provide a listing of some of the programs (those implemented in Fortran) and subroutines used in the implementation of the program DDHA (Section 5.1). The operation of each of these programs is described in Section 5.1.

```
      SUBROUTINE INIT(IG,IQ,N,M,A)
C     ************************************
C     THIS SUBROUTINE READS IN THE INITIAL
C     PROGRAM STEP SET IG AND INITIAL
C     STATE SET IQ AND ENCODES THEM
C     AS A ONE DIMENSIONAL ARRAY A
C     ************************************
      INTEGER A(1)
      INTEGER IG(100),IQ(100)
C     ************************************
C     N AS INPUT IS THE FILE WHERE IG*IQ
C     IS STORED.AS OUTPUT IT IS THE NUMBER OF
C     ELEMENTS IN IG.
C     ****************************************
      CALL FOLDING(IG,M,N,K,IQ,A,-N)
      N=K
      RETURN
      END



      SUBROUTINE FOLDING(G,M,N,K,Q,A,NS)
C     ************************************
C     FOLDING COMPUTES A ONE DIMENSIONAL ARRAY
C     A OF INTEGERS THAT CODES FOR PAIRS OF INTEGERS
C     OF ARRAYS G AND Q IF DIM(G)=N AND DIM(Q)=N
C     DIM(A)=N*M
C     ************************************
      DIMENSION IBUFF(208),X(500)
      INTEGER A(1),G(1),Q(1)
      IF(NS.LT.0) GO TO 90
      WRITE (25,10)
10    FORMAT(10X,'TYPE DIM(G),DIM(Q) IN F-FORMAT PLUS RETURN'/)
      READ(10,11) XN,XM
11    FORMAT(F12.0)
      N=XN
      M=XM
      WRITE(25,12)
12    FORMAT(10X,'TYPE ARRAY G IN F-FORMAT. AFTER EACH NUMBER
     1 TYPE "RETURN"'/)
      READ(10,11) (X(I),I=1,N)
      DO 13 I=1,N
13    G(I)=X(I)
      WRITE (25,14)
14    FORMAT(10X,'TYPE ARRAY Q INF-FORMAT. AFTER EACH NUMBER
     1 TYPE "RETURN"'/)
      READ(10,11)(X(I),I=1,M)
      DO 15 I=1,M
```

```
      ************************************************
      CONTINUATION OF SUBROUTINE FOLDING(G,M,N,K,Q,A,NS)
      ************************************************
15    Q(I)=X(I)
      GO TO 25
C     *********************************
C     NS LESS THAN ZERO, READS FROM
C     FILE -NS IN VOLUME 120
C     *********************************
90    OPEN(21,1,16,@IBUFF,'@G*Q*')
      NS=-NS
      REWIND(21)
      IF(NS-1)199,199,200
200   DO 199 I=1,NS
      SKIPFILE(21)
199   CONTINUE
      READ(21)N,(G(I),I=1,N),M,(Q(I),I=1,M)
      CLOSE(21)
25    K=AMAXO(M,N)
      IF(N.GT.M) GO TO 18
      DO 17 I=N+1,M
17    G(I)=0
      GO TO 20
18    DO 19 I=M+1,N
19    Q(I)=0
20    DO 21 I=1,K
21    A(I)=((G(I)+Q(I))**2+3*G(I)+Q(I))/2
      RETURN
      END
```