Chapter 13

Eigenbehavior: Some Algebraic Foundations of Self-Referential System Processes

13.1 Introduction

This chapter is concerned with representing organizational closure in operational terms. To this end we shall go beyond what was presented in the last chapter to construct two key notions: infinite trees of operators and solutions of equations over them. The idea of a solution of an equation over the class of infinite trees is an appropriate way to give more precise meaning to the intuitive idea of coordinations and simultaneity of interactions. The self-referential and recursive nature of a network of processes, characteristic of the autonomy of natural systems, is captured by the *invariant* behavior proper to the way the component processes are interconnected. Thus the complementary descriptions behavior/recursion (cf. Chapter 10) are represented in a nondual form. The (fixed-point) invariance of a network can be related explicitly to the underlying recursive dynamics; the component processes are seen as unfoldment of the unit's behavior.

13.2 Self-Determined Behavior: Illustrations

I propose the name *eigenbehavior* for an expression in the mathematical sense described below that is intended to represent the autonomy of some concrete system.

The name seems justified on several counts. First, the prefix "eigen" carries from German the connotation of "proper" and "self," and eigenbehavior is properly or self-determined behavior, i.e., autonomy. Second, this compound is a generalization consistent with the standard use of "eigenvalue" and "eigenvector" in linear algebra to denote certain fixed

points of linear maps. Thirdly, in at least two fields the term eigenbehavior has been proposed to denote, in particular instances, exactly what from our point of view is a solution to some system's closure. N. Jerne (1974) introduced the idea as a qualitative characterization for the moment-to-moment stable state of the totality of cellular interactions that specifies the immune network in living organisms. (We shall elaborate on this in Chapter 14.) Von Foerster's (1977) paper is entitled "Objects: tokens for eigenbehavior," and discusses the closure of the sensory-motor interactions in a nervous system, giving rise to perceptual regularities as objects. Our usage, then, not only is linguistically appropriate, but also extends previous usage to a more general systemic and mathematical content.

Even in a very general, informal sense, the notion of eigenbehavior is quite interesting. Let us consider a few illustrations of it before going into the more detailed treatment.

Eigenbehaviors can be characterized as the fixed points of certain transformations. Consider an operation α, from a domain A to itself, $\alpha: A \to A$. A fixed point for α is a value $v \in A$ such that $\alpha(v) = v$. Fixed points, in general, have several interesting properties. First, in a naive sense, a fixed point is self-referential or recursive: v says something about itself, namely, that it is invariant under the operation α. Second, fixed points are uniquely characterized with respect to all the other values taken by the operation α. Consider for example the case where α is the function $\cos: \mathrm{R} \to \mathrm{R}$. Then it is easy to verify that $x^{\nabla} = 0.739085$ [rad] is a fixed point, and in fact the only one among the continuum of values taken by cos. Third, fixed-point values can be expressed through repeated or indefinite iterations of the operations to which they are related; that is, they can be "unfolded" in terms of their defining operations. For example, we may express x^{∇} by an indefinite iteration of the operation cos, i.e., $x^{\nabla} = \cos(\cos(\cos(\cdots)))$. Note that we may disregard the value on which the iteration was initiated; it can be any number in the domain R. Now to some examples.

A rather witty illustration of such eigenbehaviors, due to von Foerster, can be described in the linguistic domain. Take the following sentence form:

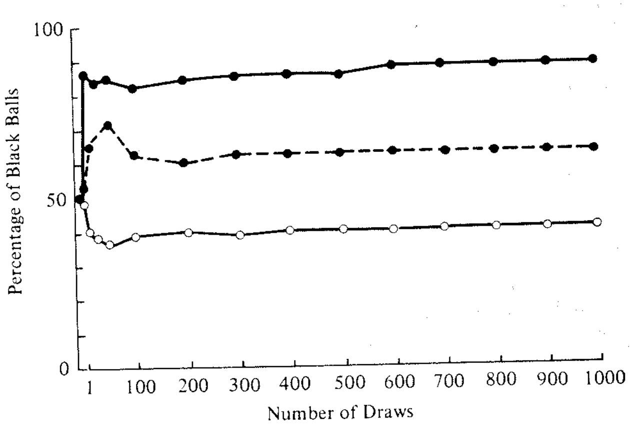
S: "This sentence has . . . letters."

Let $S(n)$ be the number of letters in S when we insert the verbal name of the number n in the empty slot. Thus $S(3) = 27$, since, "three" has 5 letters, which we add to the 22 constant letters of S. By trial and error we find that $S(33) = 33$ is the only fixed point. Only for "This sentence has thirty-three letters" does the sentence have the mentioned number of letters.

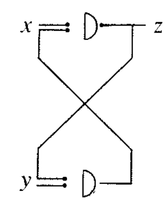
Even for a fairly simple process, the resulting eigenbehaviors can be surprisingly complex. Let me illustrate this fact. Consider an urn containing one white ball and one black ball. Let us perform the following experiment: draw one ball from the urn, at random, and whatever its color, replace it and add another ball of the same color to the urn. Repeat the above procedure many times, so that the number of balls grows very large. We then ask the question: What will be the percentage of, for example, black balls in the urn? The answer is surprising: the percentage can approach any value between 0 and 100, but in each experiment it will converge to only one stable value (Blackwell and Kendall, 1964). In other words, after an initial period of fluctuation (initial stages of approximation) the ratio will settle to a certain value and will stay close to it (eigenbehavior), although if we repeat the experiment (consider another organism of the same type) the stable value will be a different one. This experiment is illustrated in Figure 13-1. It is obvious that the outcome of the first few draws has a much more significant influence on the final value of the run than do later draws.

We may now consider a more concrete illustration, in terms of the ideas already developed in the previous chapter. Consider an electrical circuit used for computer logic, the flip-flop. One reason to choose this example is that, being used as a logical block, it can be interpreted as an indicational form (cf. Appendix B). In fact the standard diagram for the

Figure 13-1
Recursive behavior of the urn example described in the text. Three separate experiments are plotted, each up to 1000 draws. In all of them an initial stage of fluctuations is followed by a stable behavior, which differs in each case. It can be shown that there is equal probability for the behavior converging to any percentage of black balls.



flip–flop,



can be readily transposed into its corresponding eigenbehavior

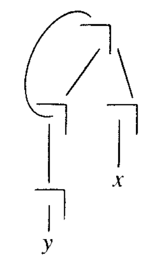
$$z = \overline{\overline{z\rceil\, \overline{y}\rceil}\rceil\, \overline{x}\rceil}, \tag{13.1}$$

or

$$z' = \overline{z}\rceil = \overline{\overline{z\rceil\, \overline{y}\rceil}\rceil\, \overline{x}\rceil}\rceil = \overline{\overline{\overline{y}\rceil\, \overline{x}\rceil}}\rceil,$$

with its tree

$z' =$



Now, z is the limit of an approximation

$$z = \bigsqcup_{n\in\omega} z_n,$$

where

$$z_1 = \overline{\overline{z_0\rceil\, \overline{y_1}\rceil}\rceil\, \overline{x_1}\rceil},$$

and, in general,

$$z_n = \overline{\overline{z_{n-1}\rceil\, \overline{y_n}\rceil}\rceil\, \overline{x_n}\rceil}. \tag{13.2}$$

Clearly

$$z_n \sqsubseteq z_{n+1}.$$

This also specifies as sequences for x and y

$$x = \bigsqcup_n x_n,$$
$$y = \bigsqcup_n y_n.$$

All of this makes sense because, in an actual flip–flop, the expression (13.1) is, of course, interpreted in time as a discrete step-by-step recursive function (13.2), for a given sequence of inputs x_i, y_i. In fact, we could have done that all along in B_∞, by interpreting z (in time) as a finite sequence, starting with some z_0, and under some finite of x_i's and y_i's, the following algebraic expression is valid (as can be easily verified by induction):

$$z_n = \overline{\overline{z_0}\rceil \alpha(n)}\rceil\, \beta(n),$$

with

$$\alpha(n) = \overline{y_1}\rceil\, \overline{y_2}\rceil \cdots \overline{y_n}\rceil,$$
$$\beta(n) = \overline{\overline{y_2}\rceil \cdots \overline{y_n}\rceil\, x_1}\rceil\, \overline{\overline{y_3}\rceil \cdots \overline{y_n}\rceil\, x_2}\rceil \cdots \overline{\overline{y_n}\rceil\, x_{n-1}}\rceil\, \overline{x_n}\rceil .$$

This is a recursive expression that algorithmically determines z_n for every n, and this is what is normally done in representing these kinds of logical circuits with feedback.

We can see, however, that this approach fits hand in glove with our approximation to an infinite expression (13.1), which embodies the self-referential quality of this reentrant circuit. The eigenbehavior represents, formally and intuitively, the basic structure of the flip–flop as a logical *design*, rather than describe it as an *ad hoc* sequential expression. The time/recursive expression shows how it can actually be operated; its reentrant forms show what it is and what it *means*.

What we see emerging from this example is that an eigenbehavior traps the intuitive idea of the global coordination or meaning of a unit, through the way in which it arises in its underlying processes. This has been standard lore in mathematical physics, where invariant transformations and fixed-point topological properties of differential dynamics are a royal road to representations of physical laws. However, these tools have been mostly concerned with numerical and differentiable representations, and there has been little development of the corresponding notions for non-numerical and informational processes. These only seem necessary when considering the phenomena proper to complex, natural systems and engineering design as well. In fact, the initial development of the ideas on continuous algebras came from the work of Scott (1971), dealing with the *semantics* of programming languages. These notions extend rather naturally to the semantics (i.e., behavior) of recursive processes in natural systems (Goguen and Varela, 1978b).

13.3 Algebras and Operator Domains

13.3.1

The next few sections are strictly concerned with the mathematical grounds necessary to represent self-referential system processes in the spirit described above. Thus the reader will be faced again with a considerable number of mathematical ideas, most of which are likely to be unfamiliar. I ask patience for this lengthy development, but I am con-

vinced that this is the sort of precision that lends some of the intuition behind this view of system's autonomy a possibility of being discussed, tested, and applied.

Four main steps follow. First, we develop some notions that are required for the representation of infinite trees: namely, operator domain, finite trees of operators, and their role in the class of algebras of operators (or Σ-algebras). Second, we present the extension of Σ-algebras to the infinite case, through order-theoretic notions and approximations. This yields the class of *continuous algebras,* and we study the role on infinite trees among them. Third, we discuss the notion of eigenbehavior as solutions of equations in continuous algebras, and we construct the set of rational (infinite) trees, which characterize recursive processes.

Throughout the presentation of these ideas, there are some difficult turns of which the reader should be forewarned, or else the technical details may seem unnecessarily complicated. The first subtle point is that, in discussing algebras of operators, we shall do so by trapping their "abstract" quality, that is, the fact that an operator name can designate many different processes in different situations. This quality of abstractness is expressed here as equivalence "up to an isomorphism" of different algebras. A second possible difficulty arises when variables are introduced into Σ-algebras and trees. The transition from simple expressions to expressions with variables seems, at first glance, simple and harmless. Thus it is surprising that when rigor is demanded, delicate steps are needed to make it come out right. In the case at hand, we end up constructing two objects [later called $T_\Sigma(X)$ and $T_{\Sigma(X)}$] which may seem mysterious. Third, the illusion that, with these tools, all our problems are gone is dispelled when we realize that the collection of infinite trees is rather unknown territory. This leads to a first classification of trees—those that we shall describe as rational—but this does not exhaust their complexity.

13.3.2

Previously (Chapter 10) we have used trees and nets to describe the connection properties of systems. But such a view does not take account of the operational capabilities of the components that are so interconnected. One step in this direction is to label each mode with a function that describes the *operation* of the associated component.

In this respect, it is important to avoid confusion between an operation and its name; for example, a careful distinction will permit us to use the same name for several operations, occurring in several situations, but having a similarity of function that it is desirable to capture. Thus, we first introduce an abstract symbol system for naming operations. The most basic quality an operation can have is the number of arguments it takes, and we include this quality in the basic notion.

Definition 13.1 *An operator domain (or signature) is a family Σ_k of disjoint sets, indexed by the natural numbers $k \in \omega$. Σ_k is the set of "operator symbols of rank k," and elements of Σ_0 are symbols for "constants" (which take no arguments).*

For an ordinary arithmetical operations, the following signature Σ would be appropriate: $\Sigma_0 = \mathbb{Z}$, the positive and negative integers; $\Sigma_1 = \{-\}$, the unary negation operator, as in the expression $-(1 + 1)$; $\Sigma_2 = \{+, \times\}$, the usual binary addition and multiplication.

An operator domain gives a basic syntax for operators, but says nothing about their semantics, that is, their meaning or interpretation. If Σ is an operator domain, then a Σ-algebra is exactly a set of elements together with a particular function for each symbol in the operator domain; that is, it gives a concrete interpretation of the abstract operation symbols. More precisely, an operation symbol $\sigma \in \Sigma_n$ is interpreted as a function $\sigma: A^n \to A$ of n arguments on a set A, and a constant symbol $\sigma \in \Sigma_0$ is interpreted as an element of A. This leads to

Definition 13.2 *Given an operator domain Σ, a Σ-algebra A is a set A, called the carrier, plus, for each $\sigma \in \Sigma_n$ with $n > 0$, a function σ_A: $A^n \to A$, and for each $\sigma \in \Sigma_0$, an element σ_A of A.*

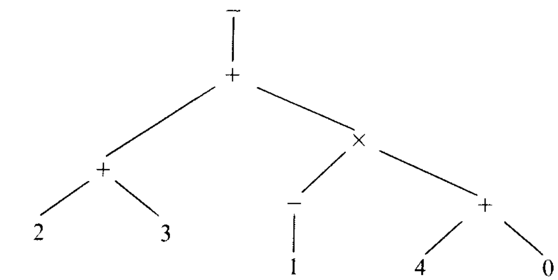
Given an operator domain Σ, we can consider *expressions* compounded from its symbols, of the general form $\sigma(t_1, \ldots, t_n)$, with σ of rank n, and each t_i either a constant symbol, or else itself a compound expression. More precisely now,

Definition 13.3 *Let Σ be an operator domain. Then the set T_Σ of all (well-formed) Σ-expressions is (recursively) the least set of expressions such that:*

1. *$\Sigma_0 \subseteq T_\Sigma$, and*
2. *if $\sigma \in \Sigma_n$, if $n > 0$, and if $t_i \in T_\Sigma$ for $i = 1, \ldots, n$, then $\sigma(t_1, \ldots, t_n) \in T_\Sigma$.*

It is possible, and suggestive, to view these Σ-expressions as trees whose nodes are labeled with symbols from Σ. Let Σ be the operator domain mentioned above. Then $-(+(+(2, 3), \times(-1, +(4, 0))))$ is a Σ-expression, which is $-((2 + 3) + ((-1) \times (4 + 0)))$ in the more usual infix notation,

and can be viewed as the tree



in which the various subexpressions correspond nicely to subtrees.

This suggests the following

Definition 13.4 *Let Σ be an operator domain. Then a Σ-tree t is a tree $\langle |t|, E, \partial_0, \partial_1, r\rangle$ (see Definition 10.1) plus a function $t: |t| \to \Sigma$ such that if the number of edges out of $a \in |t|$ is n, then $t(a) \in \Sigma_n$.*

That is, a node with n child nodes must be labeled with an operator symbol of rank n, as was the case above.

The reader may now wish to prove that there is in fact a bijective correspondence between Σ-trees and Σ-expressions. There are quite a number of equivalent infix notations for binary operators besides those mentioned in the example; there are also Polish prefix and postfix notations. For example, the above tree would be given as $-++2 \times (-1) + 40$ and $23 + (-1)\, 40 + \times + -$, respectively, in prefix and postfix notations. Again, one can establish bijective correspondences among any two of these notational systems. Moreover, the above mentioned notations far from exhaust all the possibilities.

Something is going on here: There seems to be an *abstract* underlying notion of Σ-tree or Σ-expression, which expresses the independence of the basic concept from any particular choice of how to represent it; and all representations are in some way isomorphic. This abstract quality of Σ-expressions is quite deep, and to make it more precise we begin by making T_Σ into a Σ-algebra, by defining operations as follows:

1. for $\sigma \in \Sigma_0$, $\sigma_T = \sigma$ in T_Σ, and
2. for $\sigma \in \Sigma_n$ and $t_i \in T_\Sigma$, $\sigma_T(t_1, \ldots, t_n) = \sigma(t_1, \ldots, t_n)$ in T_Σ,

where we have written σ_T for σ_{T_Σ}.

Next, we use a fundamental insight from category theory, that it is important to consider not only the "objects," but also, and perhaps more significantly, their relationships with one another, as expressed in the "structure-preserving" mappings between them. In the case of Σ-algebras, "structure-preserving" is given by

Definition 13.5 *Given an operator domain Σ and Σ-algebras A, A', a Σ-homomorphism from A to A' is a function $h: A \to A'$ such that*

1. *if $\sigma \in \Sigma_0$, then $h(\sigma_A) = \sigma_{A'}$, and*
2. *if $\sigma \in \Sigma_n$, then $h(\sigma_A(a_1, \ldots, a_n)) = \sigma_{A'}(h(a_1), \ldots, h(a_n))$.*

A Σ-homomorphism h is "structure-preserving" in the sense that if we do an operation σ in the algebra A, and then apply h, we get the same result as if we apply h to the arguments, and then do σ in A'.

We will use Σ-homomorphisms to characterize the property of being "abstractly the same as Σ-expressions," by introducing the following general notion.

Definition 13.6 *A Σ-homomorphism h is said to be an isomorphism in $\mathscr{C}$ if it has an inverse in $\mathscr{C}$, that is, a Σ-homomorphism g such that both compositions gh and hg are identities. Σ-algebras related by Σ-isomorphism are said to be isomorphic.*

For example, it is possible to make the set of all Σ-trees (see Definition 13.4) into a Σ-algebra (call it T_Σ') in such a way that the bijection between Σ-trees and Σ-expressions is actually a Σ-isomorphism between T_Σ and T_Σ'. This isomorphism makes precise the sense in which Σ-trees and Σ-expressions are "abstractly the same." Furthermore, all the other abstractly equivalent representations also give isomorphic Σ-algebras. What we now want is a more genuinely abstract way to characterize this notion. The following is the key.

Definition 13.7 *A Σ-algebra T is initial in a class $\mathscr{C}$ if there is a unique homomorphism, $h_A: T \to A$, from T to A, for all A in $\mathscr{C}$.*

A remarkable general property of initial algebras is that, if they exist, they are uniquely defined *up to isomorphism* by the class $\mathscr{C}$ on algebras within which they are initial. In algebra, the property of being "defined uniquely up to isomorphism" is said to embody the idea of abstraction; that is, initiality defines an algebra "abstractly"; this has the practical meaning of being independent of the manner of representation of elements, capturing exactly the "abstract algebraic structure" and nothing extra. The following result expresses this, and thus shows that initiality captures the notion of being "abstractly the same."

Proposition 13.8 *If T, T' are both initial in a class $\mathscr{C}$ of Σ-algebras, then T and T' are isomorphic in $\mathscr{C}$. If T' is isomorphic in $\mathscr{C}$ to an initial algebra T, then T' is also initial in $\mathscr{C}$.*

PROOF: See ADJ (1977, Proposition 1.1). □

What the above does not guarantee us is the existence of initial Σ-algebras. Let $\mathscr{A}\ell g_\Sigma$ denote the class of all Σ-algebras, together with their Σ-homorphisms. The following result was first proved by Birkhoff (1938).

Theorem 13.9 *T_Σ is initial in $\mathscr{A}\ell g_\Sigma$.*

PROOF: It will help our understanding of what is going on here to have an idea of what the unique homomorphism $h_A: T_\Sigma \to A$ looks like. If $\sigma \in \Sigma_0$, then by the definition of homomorphism, we have to have $h_A(\sigma) = \sigma_A$. Now assume that we have defined h for trees of depth $<n$, and let t be a tree of depth n. Then t is of the form $\sigma(t_1, \ldots, t_n)$, with all t_i of depth less than n. The definition of Σ-homomorphism then forces that $h_A(t) = \sigma_A(h_A(t_1), \ldots, h_A(t_n))$, and we are assuming that $h_A(t_i)$ are already well defined. Thus, $h_A(t)$ is well defined, and by induction on n, h is defined. □

This function $h_A: T_\Sigma \to A$ can be interpreted as assigning to each Σ-tree in T_Σ its "natural" interpretation in A, that is, the element that the compound Σ-expression t in fact denotes in A.

Examples 13.10

1. Let Σ be the operator domain of the example above. Then T_Σ contains trees such as that drawn above. Now let A be Z, with the operation symbols in Σ interpreted in their usual way. Then for t the tree above, $h_A(t)$ is the result of actually performing the arithmetic operations that are only symbolically indicated in t; thus $h_A(t) = 1$.
2. Let Σ be the operator domain with $\Sigma_0 = \{0\}$, $\Sigma_1 = \{s\}$, $\Sigma_k = \varnothing$, $k > 1$, where 0 is "zero" and s is "successor." Then the Σ-algebra of natural numbers ω is initial in $\mathscr{A}\ell g_\Sigma$. This provides a characterization that is different from the usual Peano postulates. MacLane and Birkhoff (1967) prove these are equivalent characterizations.

13.4 Variables and Derived Operators

For the developments to follow, we give an algebraic explication, which can be used in a Σ-term, of the concept of a "variable." We are not assuming that this is an already defined idea. In fact, this is a somewhat mysterious idea, and hope that the present discussion may contribute to its clarification.

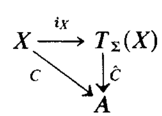
Previously, we dealt with single operators of various ranks, acting on a Σ-algebra A. We would like to be able to define compound operators, such as $x + zy(x - y)$, formed from operators in Σ and "variables" x, y, etc. The notion of a "freely generated" Σ-algebra is the key to a rigorous development of this topic.

Let $X = \{x_1, \ldots, x_n\}$ be a set of symbols disjoint from Σ, and called "variables." We first form a new signature $\Sigma(X)$ be adjoining the elements of X as new constants: $\Sigma(X)_0 = \Sigma_0 \cup X$; and $\Sigma(X)_k = \Sigma_k$, $k > 0$. Then $T_{\Sigma(X)}$ is the initial $\Sigma(X)$-algebra, and it differs from T_Σ in that its leaf nodes may carry elements of X. Because the operator symbols in $\Sigma(X)$ include those of Σ, we can think of $T_{\Sigma(X)}$ as a Σ-algebra, simply by ignoring the X part of the operator domain. More specifically, we define a new Σ-algebra, $T_\Sigma(X)$, with carrier that of $T_{\Sigma(X)}$, and with operators those named by Σ in $T_{\Sigma(X)}$.

If A is a Σ-algebra, then each element t of T_Σ has a definite interpretation $h_A(t)$ in A. However, elements of $T_{\Sigma(X)}$ or $T_\Sigma(X)$ do not have definite interpretations in A, because the elements of X do not designate definite elements of A. However, if we *assign* values in A to the variable symbols in X, using a function $C: X \to A$, then we should be able to get definite values for each element of $T_{\Sigma(X)}$; these values will, of course, in general depend upon the values assigned to the variables. In this explication, variables are constants without fixed values, but which can be assigned any desired value.

The following result shows how terms in $T_\Sigma(X)$ get values in a Σ-algebra A once the variables in X are given values in A. The advantage of using $T_\Sigma(X)$ rather than $T_{\Sigma(X)}$ is that $T_\Sigma(X)$ is a Σ-algebra, so that we can talk about Σ-homomorphisms.

Proposition 13.11 *$T_\Sigma(X)$ is the free Σ-algebra generated by X, in the sense that if $C: X \to A$ is any function mapping X into the carrier of a Σ-algebra A, then there is a unique Σ-homomorphism $\hat{C}: T_\Sigma(X) \to A$ such that following diagram commutes:*



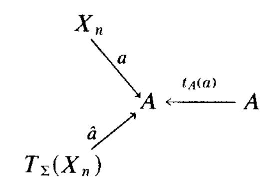
where i_X is the inclusion of X into $T_\Sigma(X)$.

PROOF: For details see ADJ (1977, Proposition 23). The following describes just the construction of $\hat{C}$ from C. Since A is a Σ-algebra, we can make A into a Σ-algebra, we can make A into a $\Sigma(X)$-algebra by letting x name $C(x)$ in A, i.e., $x_A = C(x)$. Then there is a unique $\Sigma(X)$-homomorphism $\hat{C}: T_{\Sigma(X)} \to A$. Since $\hat{C}$ is a $\Sigma(X)$-homomorphism, it is also a Σ-homomorphism. □

This result says in effect that an element t of $T_\Sigma(X)$ defines a function $t(x_1, \ldots, x_n)$ on a Σ-algebra A, since giving $x_1, \ldots, x_n$ values in A by a function $C: X \to A$ also determines a value $\hat{C}(t)$ for t in A. Thus, elements of $T_\Sigma(X)$ are themselves operators, derived from the more basic operators in Σ, and we shall call them "derived operators," or "Σ-trees in n variables."

Definition 13.12 *Let $X_n = \{x_1, \ldots, x_n\}$; let A be a Σ-algebra; let $\langle a_1, \ldots, a_n\rangle \in A^n$; and define $a: X_n \to A$ by $a(s_i) = a_i$ for $1 \le i \le n$. Then for every $t \in T_\Sigma(X_n)$ we define its corresponding derived operator on A, $t_A: A^n \to A$, by $t_A(a_1, \ldots, a_n) = \hat{a}(t)$, where $\hat{a}: t(X_n) \to A$ is the unique homorphism extending $a: X_n \to A$ guaranteed by Proposition 13.11.*

The following diagram may help visualize these relationships:



Given a Σ-tree t in n variables, we let a vary in $\hat{a}(t)$ while keeping t fixed. This amounts to "evaluating" the rank-n term t in A with variables x_i given value $a_i \in A$.

13.5 Infinite Trees

13.5.1

In this section we extend the previous ideas on Σ-algebras and Σ-trees (or Σ-terms) to the case of infinite trees (or terms). As discussed in Chapter 10, infinite trees arise as unfoldings of circular situations, and are the basis of an autonomy/control complementarity.

This extension into infinity required, however, some careful development of additional concepts. These concepts make possible the rigorous discussion of indefinite recursion. The latter requires appropriate notions of approximation and limit. The reader will have to bear with me through some rather technical material before its fruits can be seen. We shall apply some notions of order and continuity to obtain a characterization of infinite trees similar to that given for finite trees in the previous section. This material follows ADJ (1977), but is simpler, less general, more detailed, and better illustrated (Goguen and Varela, 1978b).

The fundamental concept is that of a partially ordered set or "poset." We define the order-theoretic concepts of greatest importance to us in this context.

Definition 13.13 *A poset is a set P together with a partial order $\sqsubseteq$, that is, a reflexive, antisymmetric, and transitive relation on P.*

A poset P is strict iff it has an element $\perp \in P$ such that $\perp \sqsubseteq p$ for all $p \in P$; such an element $\perp$ is called minimum or bottom for P.

An upper bound for a subset S of P is any $x \in P$ such that $a \sqsubseteq x$ for all $a \in S$. We let $(a \sqcup b)$ denote the least upper bound of $\{a, b\}$, and let $\bigsqcup S$ denote the least upper bounds (l.u.b.) of an arbitrary subset S of P.

A subset S of P is directed iff every finite subset of S has an upper bound in S.

Let $S \subseteq P$; then S is a chain iff for all $a, b \in S$, either $a \sqsubseteq b$ or $b \sqsubseteq a$. P is (ω-) chain-complete iff every (countable) chain S in P has a least upper bound in S.

Note that any two minimum elements of a poset P are in fact equal.

The natural numbers ω are a poset with the usual order. Every subset $S \subseteq \omega$ is directed, since every finite subset of numbers in S has an upper bound in S, namely the maximum of the set of numbers. Also, every subset $S \subseteq \omega$ is a chain. But ω is not chain-complete. For example, ω itself is a countable chain having no least upper bound in ω.

Let A, B be sets, and consider the set $[A \multimap B]$ of all partial functions from A to B, that is, maps for which not all a's in A have values in B; their domains may have "holes," as suggested by the notation $\multimap$. Elements of $[A \multimap B]$ correspond to subsets f of $A \times B$ satisfying the following "functional" property: If $\langle a, b \rangle \in f$ and $\langle a, b' \rangle \in f$, then $b = b'$. Then $[A \multimap B]$ is a poset with the order relation of set inclusion; least upper bounds are set unions. The latter will exist in $[A \multimap B]$ iff the set union is still a functional set; this, of course, is not always the case, but if $\langle f_i \rangle_{i \in I}$ is a chain in $[A \multimap B]$, then $\bigcup_{i \in I} f_i$ exists and is a functional set. Thus $[A \multimap B]$ is ω-chain-complete.

In line with the category-theoretic doctrine that structure-preserving maps are at least as important as the corresponding objects, we next introduce the notion of order-preserving maps for posets.

Definition 13.14 *Let P, P' be posets. Then a map f from P to P' is monotonic iff for all $p_0 \sqsubseteq p_1$ in P, $f(p_0) \sqsubseteq f(p_1)$ in P'.*

If P, P' are strict posets, then $f: P \to P'$ is strict iff $f(\perp) = \perp$.

Let P, P' be posets, and let $\langle p_i \rangle_{i \in I}$ be an ω-chain in P. Then $f: P \to P'$ is ω-chain-continuous iff

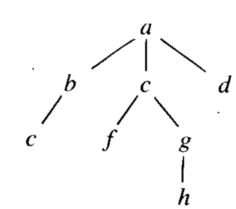
$$f(\bigsqcup_{i \in I} p_i) = \bigsqcup_{i \in I} f(p_i) \qquad \text{in } P'.$$

This last definition says that a map is continuous iff the same value results from taking the least upper bound of a chain and then looking at its image, as from mapping each member of the chain and then taking the least upper bound of the images. This notion of continuity is reminiscent of the one found in elementary calculus.

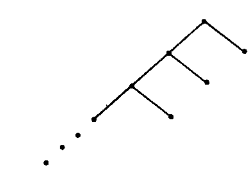
13.5.2

We are interested in putting a partial orders on sets of Σ-trees. A clue to how to do this is provided by our example above: If we can make a set of Σ-trees into a set of partial functions, of the form $[A \multimap B]$, then the set will have a natural partial order; it also seems reasonable to guess that $B = \Sigma$ will work. But it is not clear what A should be. Here we use an elegant representation of nodes by strings of natural numbers. The basic idea is that the string shall encode the sequence of choices of branches required to get from the root to the node in question. Thus, the root is represented by the empty string λ. An $(n + 1)$st child of the root is represented by the string consisting of the single integer $n + 1$; in particular, the first child is represented by 0, and the second by 1. More generally, if u is a string of non-negative integers, representing a node, then the $(n + 1)$st child of u is represented by the string un, consisting of u followed by n. The set of all possible node representations is then the set ω^* of all finite strings of non-negative integers; this is the set A we wanted above.

Before going on to Σ-trees and formal definitions, let us see how this encoding of nodes to strings works on simple examples. Consider the tree



in which a, b, c, d, e, f, g, h are the names of the nodes. These are represented by the following strings (in the same order): λ, 0, 1, 2, 00, 10, 11, 110. One advantage of this approach is that it extends to infinite trees without difficulty. For example, the infinite tree structure



has as its set of node representations λ, 0, 1, 00, 01, 000, 001, . . . ; to be precise, it is $\{0^n \mid n \in \omega\} \cup \{0^n 1 \mid n \in \omega\}$.

Clearly, not any set of strings can be the set of representations of the nodes of a tree. Those sets that can be are captured in the following. At the same time, we show how to handle Σ-trees.

Definition 13.15 *A full tree domain is a subset D of ω^* such that, for all $u \in \omega^*$ and $n \in \omega$,*

1. *$un \in D$ implies $u \in D$,*
2. *$un \in D$ implies $ui \in D$ for all $i \in \omega$ with $i < n$.*

Let $\langle \Sigma_n \rangle_{n\in\omega}$ be an operator domain, and let Σ denote the set $\bigcup_n \Sigma_n$. Then a full Σ-tree is a partial function $t: \omega^* \nrightarrow \Sigma$ such that the domain of definition of t is a tree domain D such that

3. *if $u \in D$ but $ui \notin D$ for all $i \in \omega$, then $t(u) \in \Sigma_0$,*
4. *if $ui \in D$ and $i \in \omega$, then $t(u) \in \Sigma_n$ for some $n > i$.*

We shall say that a tree t is finite iff its domain is finite. Let FF_Σ denote the set of all finite full Σ-trees, and let FT_Σ denote the set of all full Σ-trees, finite or not.

Now the set FF_Σ of all full finite Σ-trees can be given operations $\sigma_F: FF_\Sigma{}^n \to FF_\Sigma$ for each $\sigma \in \Sigma_n$ in a way analogous to those given earlier for T_Σ, and it can be shown that the resulting Σ-algebra is an initial Σ-algebra. Therefore it is isomorphic to T_Σ, and we have another example of a different representation of the same structure. Notice that the carrier of FF_Σ is the set of all finite elements of $[\omega^* \nrightarrow \Sigma]$ satisfying conditions 1–4 above. We shall hereafter identify FF_Σ and T_Σ, both as Σ-algebras and as sets.

There is, however, a problem with our plan to use this approach to get an order structure on Σ-trees: the order relation on full Σ-trees is not very interesting. In fact, we have the following

Proposition 13.16 *For $t, t' \in [\omega^* \nrightarrow \Sigma]$, define $t \sqsubseteq t'$ to mean that, as sets of ordered pairs (that is, as subsets of $\omega^* \times \Sigma$), t is a subset of t'. Then, if t, t' are full Σ-trees and $t \sqsubseteq t'$, either $t = t'$ or $t = \varnothing$, the empty tree.*

PROOF: If t is a full Σ-tree, D is its domain, and $u \in D$, then $t(u) \in \Sigma_n$ iff u has exactly n children, namely, $u0, u1, \ldots, u(n-1)$.

Let D, D' be the domains of t, t' and assume $t \neq \varnothing$, $t \sqsubseteq t'$, and $t \neq t'$. Then there is some $u \in D' - D$. Write $u = vw$, choosing v of maximum possible length such that $v \in D$ (this is possible, because at worst $v = \lambda$, and both v and w are uniquely determined by giving the length of v). By conditions 1 and 2 of Definition 13.15 (and induction), $v \in D'$ and $t'(v) \in \Sigma_n$ with $n > 0$. By condition 3 of Definition 13.15, $t(v) \in \Sigma_0$. But $t \sqsubseteq t'$ implies $t(v) = t'(v)$. We saw that this is impossible, so the assumption that $t \neq t'$ must have been wrong. □

What we really want are finite approximations to the infinite Σ-trees. These are easy to obtain, if we relax the requirement that if $t(u) \in \Sigma_n$ then u has exactly n children, by letting some of the child nodes be "undefined." The following shows how we do this.

Definition 13.17 *A (partial) tree domain is a subset D of ω^* such that for all $u \in \omega^*$ and $n \in \omega$,*

1. *$un \in D$ implies $u \in D$.*

Let $\langle \Sigma_n \rangle_{n\in\omega}$ be an operator domain. Then a (partial) Σ-tree is a partial function $t: \omega^* \nrightarrow \Sigma$ such that the domain of definition D of t is a partial tree domain, and

2. *if $ui \in D$ and $i \in \omega$, then $t(u) \in \Sigma_n$ for some $n > i$.*

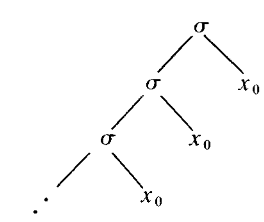
(Thus, a partial tree satisfies conditions 1 and 4 of Definition 13.15, but not necessarily 2 and 3.) If t, t' are partial Σ-trees, then $t \sqsubseteq t'$ iff $t \subseteq t'$ as sets of ordered pairs). Let CT_Σ denote the set of all partial Σ-trees (both finite and infinite).

The following table should help in keeping track of the notation for various kinds of Σ-trees:

	Full	Partial (or full)
Finite	$FF_\Sigma \cong T_\Sigma$	F_Σ
Finite or infinite	FT_Σ	CT_Σ

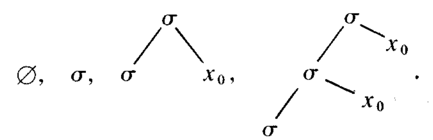
Note that $F_\Sigma \subseteq CT_\Sigma$. Hereafter, we shall feel free to drop the word "partial" and refer to elements of CT_Σ and F_Σ as "Σ-trees." To avoid confusion, elements of T_Σ and FT_Σ will be referred to as "full Σ-trees."

We illustrate that the ordering relation $\sqsubseteq$ on CT_Σ is nontrivial. Let t be the following full Σ-tree:



where $\sigma \in \Sigma_2$ and $x_0 \in \Sigma_0$. This tree has as its domain D that given in the previous example. We now construct a sequence $t^{(0)}, t^{(1)}, t^{(2)}, \ldots$ of finite partial Σ-tree approximations to t: let $D^{(n)} = \{u \in D \mid u \text{ has length } < n\}$, and let $t^{(n)}$ be the restriction of t to $D^{(n)}$.

For example, $D^{(0)} = \varnothing$, $D^{(1)} = \{\lambda\}$, $D^{(2)} = \{\lambda, 0, 1\}$, $D^{(3)} = \{\lambda, 0, 1, 00, 01\}$; and $t^{(0)}, t^{(1)}, t^{(2)}, t^{(3)}$ look like



Clearly $t^{(n)} \sqsubseteq t^{(n+1)} \sqsubseteq t$ for all $n \in \omega$. Moreover, $\bigcup_n t^{(n)} = t$.

That this situation generalizes nicely, is shown by the following

Proposition 13.18 CT_Σ *is a chain-complete poset.*

PROOF: We recall from a previous example that chains of partial functions have least upper bounds that are partial functions. Now, let $\langle t_i \rangle_{i \in I}$ be a chain of partial functions $\omega^* \nrightarrow \Sigma$, each satisfying conditions 1 and 2 of Definition 13.17. Then it is not hard to show that the union $\bigcup_{i \in I} t_i$ also satisfies 1 and 2, and is therefore in CT_Σ. Therefore, it is the least upper bound of $\langle t_i \rangle_{i \in I}$. □

In particular, CT_Σ is ω-chain-complete, by restricting to chains $\langle t_i \rangle_{i \in I}$.

Proposition 13.19 *Let $t \in CT_\Sigma$, and let D be the domain of t. Let $D^{(n)} = \{u \in D \mid (\text{length of } u) < n\}$, and let $t^{(n)}$ be the restriction of t to $D^{(n)}$. Then $\langle t^{(n)} \rangle_{n \in \omega}$ is an ω-chain of finite Σ-trees with least upper bound t.*

13.6 Continuous Algebras

The set CT_Σ of all finite and infinite (partial) Σ-trees can be given the structure of a Σ-algebra, in very much the same way as was done earlier for the finite full Σ-trees. This makes CT_Σ into a kind of ordered algebra and leads toward the characterization of the algebra of Σ-trees as initial in various classes of ordered algebras.

Definition 13.20 *Let Σ be an operator domain. We make CT_Σ into a Σ-algebra as follows, making quite explicit use of the ordered-pair representation of partial functions:*

1. *for $\sigma \in \Sigma_0$, let $\sigma_{CT} = \{\langle \lambda, 0 \rangle\}$;*
2. *for $\sigma \in \Sigma_n$, $n > 0$, and $t_1, \ldots, t_n \in CT_\Sigma$, let $\sigma_{CT}(t_1, \ldots, t_n) = \{\langle \lambda, \sigma \rangle\} \cup \bigcup_{i<n} \{\langle iu, \sigma' \rangle \mid \langle u, \sigma' \rangle \in t_{i+1}\}$,*

where we have written σ_{CT} rather than σ_{CT_Σ}.

Informally, a compound expression $\sigma(t_1, \ldots, t_n)$ is obtained by taking a node labeled σ as root, and attaching the roots of the trees $t_1, \ldots, t_n$ below it (as children), in order.

Now, CT_Σ already has a partial ordering. We make use of the algebraic operations given above to provide a characterization of the order structure of CT_Σ analogous to that provided by Proposition 13.16 for FT_Σ.

Proposition 13.21 *For $t, t' \in CT_\Sigma$, $t \sqsubseteq t'$ iff: $t = \varnothing$; or $t = t' = \sigma_{CT}$ for $\sigma \in \Sigma_0$; or there is some $\sigma \in \Sigma_n$ with $n > 0$, and $t_1, \ldots, t_n, t_1', \ldots, t_n' \in CT_\Sigma$ such that $t = \sigma_{CT}(t_1, \ldots, t_n)$, $t' = \sigma_{CT}(t_1', \ldots, t_n')$, and $t_i \sqsubseteq t_i'$ for $1 \le i \le n$.*

PROOF: See ADJ (1977). □

Not only is CT_Σ an ω-complete poset and a Σ-algebra, but these structures are related, in the following particularly felicitious way.

Proposition 13.22 *For each $\sigma \in \Sigma$, the operation σ_{CT} on CT is ω-chain-continuous.*

PROOF: Let $\sigma \in \Sigma_n$. The result is trivial if $n = 0$, so we may assume $n > 0$.

Let $\langle t_{ij} \rangle_{j \in \omega}$ be ω-chains in CT_Σ for $i = 1, \ldots, n$. Let $t_i = \bigsqcup_{j \in \omega} t_{ij}$; these l.u.b.'s exist by Proposition 13.19, and in fact, they are set unions.

Now, it is "well known," for the n-fold Cartesian-product poset $CT_\Sigma \times \cdots \times CT_\Sigma$ ordered by $\langle t_1, \ldots, t_n \rangle \sqsubseteq \langle t', \ldots, t_n' \rangle$ iff $t_i \sqsubseteq t_1'$ for $i = 1, \ldots, n$, that l.u.b.'s can be computed componentwise; that is, for $t_{ij} \in CT_\Sigma$ for $i = 1, \ldots, n$ and $j \in \omega$,

$$\bigsqcup_{j \in \omega} \langle t_{1j}, \ldots, t_{nj} \rangle = \langle \bigsqcup_{j \in \omega} t_{1j}, \ldots, \bigsqcup_{j \in \omega} t_{nj} \rangle$$

(which is $\langle t_1, \ldots, t_n \rangle$).

What we want to show is that

$$\sigma_{CT}(\bigsqcup_{j \in \omega} t_{1j}, \ldots, \bigsqcup_{j \in \omega} t_{nj}) = \bigsqcup_{j \in \omega} \sigma_{CT}(t_{1j}, \ldots, t_{nj}).$$

So let us calculate, using Definition 13.20.

$$\begin{aligned}
\sigma_{CT}(t_1, \ldots, t_n) &= \{\langle \lambda, \sigma \rangle\} \cup \bigcup_{i<n} \{\langle iu, \sigma' \rangle \mid \langle u, \sigma' \rangle \in \bigsqcup_{j \in \omega} t_{i+1,j}\} \\
&= \{\langle \lambda, \sigma \rangle\} \cup \bigcup_{i<n} (\bigcup_{j \in \omega} \{\langle iu, \sigma' \rangle \mid \langle u, \sigma' \rangle \in t_{i+1,j}\}) \\
&= \bigcup_{j \in \omega} (\{\langle \lambda, \sigma \rangle\} \cup \bigcup_{i<n} \{\langle iu, \sigma' \rangle \mid \langle u, \sigma' \rangle \in t_{i+1,j}\}) \\
&= \bigsqcup_{j \in \omega} \sigma_{CT}(t_{1j}, \ldots, t_{nj}).
\end{aligned}$$

□

Definition 13.23 *An ordered Σ-algebra is a Σ-algebra whose carrier is a strict poset, and whose operations are monotonic. A homomorphism* *of an ordered Σ-algebra is a strict monotonic Σ-homomorphism. Let $\mathcal{P}a\ell g_\Sigma$ denote the class of all ordered Σ-algebras, together with all strict monotonic homomorphisms among them.*

An ω-continuous Σ-algebra is a Σ-algebra whose carrier is a strict ω-complete poset whose operations are ω-continuous. A homomorphism of ω-continuous Σ-algebras is a strict ω-continuous Σ-homomorphism. Let $\omega a\ell g_\Sigma$ denote the class of all ω-continuous Σ-algebras, together with all strict ω-continuous Σ-homomorphisms among them.

We have shown that CT_Σ is an ω-continuous Σ-algebra, and thus, an ordered Σ-algebra. The result we are aiming for is that CT_Σ is initial in $\omega a\ell g_\Sigma$. The proof uses the following two results, the first of which is certainly of independent interest.

Proposition 13.24 *F_Σ is initial in $\mathcal{P}a\ell g_\Sigma$.*

PROOF: Let $\Sigma(\bot)$ denote the signature Σ enriched by the new constant symbol ⊥. Now, we can make any ordered Σ-algebra A into a $\Sigma(\bot)$-algebra, by letting ⊥ in $\Sigma(\bot)$ denote ⊥ in A, which exists because A is strict. Also, note that a strict Σ-homomorphism is the same thing as a $\Sigma(\bot)$-homomorphism.

The reader may want to verify the following lemma, upon which this proof relies: F_Σ, as a $\Sigma(\bot)$-algebra, is isomorphic to $T_{\Sigma(\bot)}$. Then, for any ordered Σ-algebra A, there is a unique $\Sigma(\bot)$-homomorphism h_a: $F_\Sigma \to A$, and we shall be done if we can show that it is monotone.

Let $t \sqsubseteq t'$ in F_Σ. Since $F_\Sigma \subseteq CT_\Sigma$, we can apply Proposition 13.22 to get that either: (1) $t = \bot$, or (2) $t = t' = \sigma_F$ for $\sigma \in \Sigma_0$, or (3) $t = \sigma_F(t_1, \ldots, t_n)$ and $t' = \sigma_F(t_1', \ldots, t_n')$ with $t_i \sqsubseteq t_i'$ (for $i = 1, \ldots, n$) and $\sigma \in \Sigma_n$, noting that each t_i and t_i' must be in F_Σ, not just in CT_Σ.

In case (1), $h_A(t) = \bot$, since h_A is strict, so certainly $h_A(t) \sqsubseteq h_A(t')$.

In case (2), obviously $h_A(t) \subseteq h_A(t')$ since $h_A(t) = h_A(t')$.

Case (3) is the interesting one, and the proof proceeds by induction on the cardinality of the domain of definition of t. Cases (1) and (2) above are in fact the basis of the induction. For the inductive step, we assume that $h_A(t_i) \sqsubseteq h_A(t_i')$, and calculate

$$\begin{aligned}
h_A(t) &= h_A(\sigma_{\mathrm{CT}}(t_1, \ldots, t_n)) && \text{[form of } t \text{ from (3)]}\\
&= \sigma_A(h_A(t_1), \ldots, h_A(t_n)) && [h_A \text{ is a homomorphism}]\\
&\sqsubseteq \sigma_A(h_A(t_1'), \ldots, h_A(t_n')) && [\sigma_A \text{ is monotone}]\\
&= h_A(\sigma_F(t_1', \ldots, t_n')) && [h_A \text{ is a homomorphism}]\\
&= h_A(t'). && \text{[form of } t' \text{ from (3)]}
\end{aligned}$$

Thus $h_A(t) \sqsubseteq h_A(t')$. □

Proposition 13.25 *The operations of F_Σ are ω-chain-continuous.*

PROOF: The proof of Proposition 13.24 goes through when restricted to trees with finite domains. □

Now the main result.

Theorem 13.26 *CT_Σ is initial in $\omega a\ell g_\Sigma$.*

PROOF: The proof is based on the fact that $\omega a\ell g_\Sigma \subseteq \mathcal{P}a\ell g_\Sigma$, so that for any A in $\omega a\ell g_\Sigma$, there is a unique strict monotone Σ-homomorphism $h_A: F_\Sigma \to A$. The work of the proof is to extend this to an ω-continuous Σ-homomorphism $\bar{h}_A: \mathrm{CT}_\Sigma \to A$, using the approximation suggested by Proposition 13.21: For $t \in \mathrm{CT}_\Sigma$, we have $t = \bigsqcup_{n\in\omega} t^{(n)}$, with each $t^{(n)} \in F_\Sigma$; we then define (when no confusion will arise, we write $\bigsqcup_n$ for $\bigsqcup_{n\in\omega}$)

$$\bar{h}_A(t) = \bigsqcup_n h_A(t^{(n)}),$$

knowing that this l.u.b. exists because A is ω-continuous. It remains to show that (1) the extension is unique, (2) is ω-continuous, and (3) is a Σ-homomorphism.

1. Suppose $h': \mathrm{CT}_\Sigma \to A$ extends $h_A: F \to A$ and is ω-continuous. Then
$$h'(t) = h'(\bigsqcup_n t^{(n)}) = \bigsqcup_n h'(t^{(n)}) = \bigsqcup_n h_A(t^{(n)}) = \bar{h}_A(t).$$

2. We first show that $\bar{h}_A$ is monotonic. Let $t_0 \sqsubseteq t_1$ in CT_Σ. Then $t_0^{(n)} \sqsubseteq t_1^{(n)}$ for all $n \in \omega$, so that $h_A(t_0^{(n)}) \sqsubseteq h_A(t_1^{(n)})$ by monotonicity of h_A; therefore
$$\bar{h}_A(t_0) = \bigsqcup_n h_A(t_0^{(n)}) \sqsubseteq \bigsqcup_n h_A(t_1^{(n)}) = \bar{h}_A(t_1),$$
as desired.

   Now assume that $t = \bigsqcup_i t_i$, for $\langle t_i \rangle_{i\in\omega}$ an ω-chain in CT_Σ. We want to show that $\bigsqcup_i \bar{h}_A(t_i) = \bar{h}_A(\bigsqcup_i t_i) = \bar{h}_A(t)$.

   The key lemma is the following: For each $n \in \omega$, there is some $j \in \omega$ such that $t^{(n)} \sqsubseteq t_i$. To show this, it suffices to show that for any sets A, B, if $t' \in [A \Rightarrow B]$ is finite, and if $t' \sqsubseteq \bigsqcup_i t_i$ for $\langle t_i \rangle_{i\in\omega}$ a chain in $[A \Rightarrow B]$, then there is some $j \in \omega$ such that $t' \sqsubseteq t_j$.

   Now let $b \in \mathrm{CT}_\Sigma$ be an upper bound of the chain $\langle \bar{h}_A(t_i) \rangle_{i\in\omega}$, i.e., $\bar{h}_{A(t)} \sqsubseteq b$ for all $i \in \omega$. Then (by the lemma of the above paragraph), for each $n \in \omega$, there is some $j \in \omega$ such that $\bar{h}_A(t^{(n)}) \sqsubseteq \bar{h}_A(t_j) \sqsubseteq b$. Therefore, $\bigsqcup_n \bar{h}_A(t^{(n)}) = \bar{h}_A(t) \sqsubseteq b$. It now follows that $\bar{h}_A(t) = \bigsqcup_i \bar{h}_A(t_i)$; i.e., that $\bar{h}_A$ is ω-continuous, as desired.

3. We now show that $\bar{h}_A$ is a Σ-homomorphism. First, observe that for each $\sigma \in \Sigma_n$, $t_i \in \mathrm{CT}_\Sigma$ for $i = 1, \ldots, n$, and $k > 0$.
$$\sigma_{\mathrm{CT}}(t_1, \ldots, t_n)^{(k)} = \sigma_{\mathrm{CT}}(t_1^{(k-1)}, \ldots, t_n^{(k-1)}),$$

while $t^{(0)} = \perp$. Now let us compute:

$$\begin{aligned}
&\bar h_A(\sigma_{CT}(t_1, \ldots, t_n)) \\
&= \bigsqcup_k h_A(\sigma_{CT}(t_1, \ldots, t_n)) && [\text{definition of } \bar h_A] \\
&= \bigsqcup_k h_A(\sigma_{CT}(t_1^{(k-1)}, \ldots, t_n^{(k-1)})) && [\text{above observation}] \\
&= \bigsqcup_k \sigma_A(h_A(t_1^{(k-1)}), \ldots, h_A(t_n^{(k-1)})) && [h_A \text{ is a } \Sigma\text{-homomorphism}] \\
&= \sigma_A(\bigsqcup_k h_A(t_1^{(k-1)}), \ldots, \bigsqcup_k h_A(t_n^{(k-1)})) && [\sigma_A \text{ is } \omega\text{-continuous}] \\
&= \sigma_A(\bar h_A(t_1), \ldots, h_A(t_n)) && [\text{definition of } \bar h_A]
\end{aligned}$$

(where some subscripts k range over $k > 0$ rather than $k \in \omega$). □

This completes our general discussion of infinite trees and initial continuous algebras.

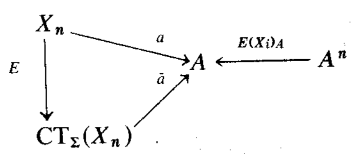
13.7 Equations and Solutions

We are now ready to give the definitions of solutions of an equation over a continuous algebra. This will enable us to formalize the idea of an eigenbehavior, discussed at the outset of this chapter in a general way.

Definition 13.27 *A system of n equations in CT_Σ is a function E: $X_n \to CT_\Sigma(X_n)$. We write $x_i = E(x_i)$ as the ith equation.*

For any A in $\omega\mathcal{alg}_\Sigma$, $E_A: A^n \to A^n$ is the derived operator of E over A, $E_A = \langle E(x_1)_A, \ldots, E(x_n)_A\rangle: A^n \to A^n$.

This definition can be represented in the diagram



so that $(E_A(a))_i = E(x_i)_A(a) = \bar a(E(x_i))$.

Proposition 13.28 *Let E be a system of n equations in CT_Σ, A in $\omega\mathcal{alg}_\Sigma$. Then E_A has a minimum fixed point $|E_A| \in A^n$, called the* solution *of E over A, or the* eigenbehavior *of E over A.*

PROOF: Define $|E_A| = \bigsqcup_{k\in\omega} E_A{}^k(\perp, \ldots, \perp)$. Then $|E_A|$ is a fixed point, since

$$\begin{aligned}
E_A(|E_A|) &= E_A(\bigsqcup_{k\in\omega} E_A{}^k(\perp, \ldots, \perp)) \\
&= E_A(\bigsqcup_{k\in\omega} E_A{}^k(x_i)(\perp), \ldots, \bigsqcup_{k\in\omega} E_A{}^k(x_n)(\perp)) \\
&= (\bigsqcup_{k\in\omega} E_A{}^{k+1}(x_1)(\perp), \ldots, \bigsqcup_{k\in\omega} E_A{}^{k+1}(x_n)(\perp)) \\
&= |E_A|.
\end{aligned}$$

Consider now $a \in A^n$ such that $E_A(a) = a$. Then clearly $|E_A|^{(0)} \sqsubseteq a^{(0)}$. Assume, for induction, that $|E_k|^{(k)} \sqsubseteq a^{(k)}$. Then, since E_A is monotonic,

$$E_A(|E_A|^{(k)}) = |E_A|^{(k+1)} \sqsubseteq E_A(a^{(k)}) = a^{(k+1)}.$$

Thus, $|E_A| \sqsubseteq a$, and $|E_A|$ is the minimum fixed point. □

This proposition shows how to construct a fixed point of E through the indefinite iteration of the trees forming the system of equations.

Call $|E|$ the solution for the case $A = CT_\Sigma$, and write $|E|_A = h_A{}^n(|E|)$. We now show that we can either solve an equation and then interpret it, or first interpret and then solve.

Proposition 13.29 *For any system $E: X_n \to CT_\Sigma(X_n)$ of equations and any $A \in \omega\mathcal{alg}_\Sigma$, $|E|_A = |E_A|$.*

PROOF: By induction on k it is clear that

$$h_A{}^n(E^k(\perp, \ldots, \perp)) = E^k(\perp_A, \ldots, \perp_A).$$

Then

$$\begin{aligned}
|E|_A &= h_A{}^n(|E|) \\
&= h_A{}^n(\bigsqcup_{k\in\omega} E^k(\perp, \ldots, \perp)) \\
&= \bigsqcup_{k\in\omega} h_A{}^n(E^k(\perp, \ldots, \perp)) && (\text{continuity of } h) \\
&= \bigsqcup_{k\in\omega} E_A{}^k(h_A(\perp), \ldots, h_A(\perp)) \\
&= \bigsqcup_{k\in\omega} E_A{}^k(\perp_A, \ldots, \perp_A) \\
&= |E_A|.
\end{aligned}$$

□

We have then that every system of equations has an eigenbehavior over CT_Σ; conversely, this is a way in which we could hope to characterize infinite trees of CT. Could we not associate with an infinite tree the equation(s) for which it is a solution? We would like such equational elements of CT_Σ to be well behaved, in the sense of being describable and having adequate composition properties. But there is no assurance that this is the case: The problem of dealing with equational elements of CT_Σ is quite complex (ADJ, 1978). We can, however, say something more precise about a *small* part of CT_Σ, namely those infinite trees that are solutions for *finite* systems of equations, that is, systems E such that $E: X_n \to F_\Sigma(X_n)$. An ordered algebra is said to be *equationally complete* if every finite system of equations has a solution over A.

Let R_Σ denote the set of equational elements of CT that are solutions for finite systems, i.e., the collection of eigenbehaviors

$$R_\Sigma = \{|E|_i \mid E: X_n \to F_\Sigma(X_n),\ n > 0,\ 1 \le i \le n\}.$$

The reason for this definition of equational completeness and the use of the "R" in R_Σ is the following characterization of the trees in R_Σ, which we state without proof (see ADJ, 1977, Propositions 5.3, 5.4).

Proposition 13.30 *If* $t \in R_\Sigma$, *then for each* $\sigma \in \Sigma_n$, $t^{-1}(\sigma) \subseteq \omega^*$ *is a regular subset of* $\{0, 1, \ldots, k\}^*$ *for some* k.

Thus the elements of R_Σ can be described and compared by means of some computable procedures. This is, of course, *not* the case for all the other elements in CT_Σ: Some infinite trees might not even be finitely or even recursively describable, and we have no idea how they behave under composition, quotient, and so on.

By contrast the elements of R_Σ are very well behaved: R_Σ is a subalgebra of CT_Σ (ADJ, 1977, Proposition 5.5). (Accordingly Scott calls elements in R_Σ "algebraic," and elements in $CT_\Sigma - R_\Sigma$ "transcendental.") For our present purposes we need only be concerned with the construction of R_Σ as an equationally complete subalgebra, since finite systems of equations are certainly the ones that are needed in most (if not all) the concrete applications. This is so to the extent that equations embody the ways in which the system components interconnect. We may assume this situation to be always captured in a finite description (i.e., in trees of F_Σ).

13.8 Reflexive Domains

Let us pause for a moment to reconsider what these algebraic developments mean in the broader context of the investigation proposed in these pages. Paying attention to the autonomy of natural systems led us to the closure thesis—that is, to consider the complementarity between the recursive underlying dynamics of a unity, and the way in which such dynamics generates a coherent pattern, a behavior of a unity affording a criterion of distinction. In order to carry this characterization one step further, we decided to make precise what we mean by complementarity, and by a coherent behavior and its underlying processes. That is the spirit of the notion of organizational closure, and of complementarity as adjunction, developed so far. Further precision of these ideas hinges upon the construction of appropriate calculi, where elements or operands are on the same descriptive level with operators or processes, and where the products of processes become effectively interrelated with the processes that generate them.

Let us formulate these notions more formally thus: Consider a descriptive domain of elements D of some kind (stable levels of reactants, coherent pattern of behavior, meaning of a conversation, and so on). By the closure thesis, these belong to an autonomous system if they arise

out of processes acting on the very same elements, that is, some appropriate class of operations or processes, which we may denote $[D \to D]$. We need to keep the distinction between elements (criteria of distinction) and processes (underlying dynamics), but in such a way that they are effectively related, that is, in such a way that they are seen as the same, except for the means we choose to observe them, in a star fashion. One way of formulating this complementarity is to demand a correspondence

$$D \overset{R}{\leftrightarrow} [D \to D]. \tag{13.3}$$

When R is an isomorphism, we call D a *reflexive domain*. It can be understood as a descriptive realm which can operate on itself (act on itself).

Now, functions or operations that can operate on themselves have been a headache in mathematics for a long time. If we simply ask whether (13.3) is true, in general, for various kinds of D's and of functions on the D's, the answer is *no*. Such reflexive domains cannot exist without inconsistencies. However, the condition (13.3) becomes possible if we *restrict* the kinds of domains and their operations (see Appendix B). We started this characterization on the simplest possible grounds: those of indicational forms. We succeeded in expressing pattern/dynamics in an explicit form. In order to carry the overall distinction into diversified operations, we presented the development of continuous algebras, where a special kind of descriptive domain (i.e., a continuous algebra) and special operations in them (i.e., continuous) could yield a correspondence as well. In fact, under these restrictions, we had

$$CT_\Sigma \underset{E}{\overset{S}{\leftrightarrows}} [CT_\Sigma \to CT_\Sigma].$$

S relates equations to their eigenbehavior (minimum fixed-point solution). E relates infinite trees to the equations of which they are a solution. In this case, the correspondence (13.3) is not an isomorphism since E is a one-to-many map. Thus CT_Σ is close to, but not identical with, a reflexive domain.

So far, this approach has been used in detail only in the semantics of programming languages, where it originated, as we have said, with the work of Scott. The motivation for Scott's work was to consider the relation between the meaning of a program (i.e., its criteria of distinction), and its computational behavior (i.e., its underlying closure). In this sense, a program is looked at as an autonomous object, as a *text*. This is, of course, not to say that the computer itself is looked as an autonomous object: We are talking about the coherence of recursive programs. These are not so distant from other texts, proper to natural languages, that arise as coherent objects (cf. Chapter 16; Becker, 1977; Linde, 1978). In the present interpretation, Scott's work, as elaborated by the theory of continuous algebras, means that this insight into the coherence of a programming text *can be generalized to the coherence of other autonomous units, providing us with precise formal tools to represent them.*

To be sure, these algebraic foundations have limitations, but they still contain a large class of possible models. For each particular system under study it is necessary to specify in detail which operator domain is to be considered, and what is its order structure. Once this is established, all the results from the theory of continuous algebras become available, since our treatment was "abstract" through the notion of initiality. In other words, this means that we begin to have available a range of mathematical tools, beyond those of differential dynamics (cf. Section 13.10), within which we can include any process whatsoever that can be made precise enough to define an operator domain satisfying the appropriate restrictions of order and continuity. I hasten to warn the reader that beyond the cases of text coherence, in programming languages (e.g., Stoy, 1977) and planning discourse (Linde and Goguen, 1978), this theory has *not* yet been applied in any detail. The ground is entirely open. In the sections that follow I shall try to give a glimpse of the flavor such applications can have, without pretending to be exhaustive.

13.9 Indicational Reentry Revisited

We can now deal more adequately with the issue raised in Chapter 12, in relation to infinite indicational expressions. We gave there an informal construction the class B_∞ of continuous forms; we shall briefly review it here, in a more rigorous form. At the same time it will serve as an exercise for in the applications of continuous algebras just presented.

Let Σ be the following operator domain: $\Sigma_0 = (1, 0)$; $\Sigma_1 = \{\sigma_1\}$; $\Sigma_2 = \{\sigma_2\}$; $\Sigma_k = \varnothing$, $k > 2$. Let B denote the set of forms in the indicational arithmetic formed by crossing and containment of the primary values *marked* ($\urcorner$) and *unmarked* ( ). Thus B is a collection of trees formed out of the carrier {$\urcorner$, , cross, containment}. Now make B into a Σ-algebra thus:

1. for $\sigma \in \Sigma_0$ let $1_B = \urcorner$, $0_B =$;
2. for $\sigma \in \Sigma_1$, and $t \in T_\Sigma$ let

$$\sigma_1(t) = \overline{t}\urcorner = \text{cross}(t);$$

3. for $\sigma_2 \in \Sigma_2$, and $t, t' \in T_\Sigma$ let

$$\sigma_2(t, t') = tt' = \text{containment}(t, t').$$

Consider now the initial Σ-algebra T_Σ. There is a unique homomorphism

$$\text{ind}: T_\Sigma \to B$$

assigning to each tree in T_Σ an interpretation as an indicational form. For instance,

$$\text{ind}[\sigma_2(\sigma_1(0), \sigma_1(1)] = \urcorner\ \urcorner = \text{ind}[\sigma_2(1, 1)].$$

Expressions in the primary indicational algebra are easily obtained by interpreting derived operators in $T_\Sigma(X_n)$, where $X_n = \{X_1, \ldots, X_n\}$ is a set of variables. So far we have simply redone the calculus of indication in the light of Σ-algebras.

The key to extending indicational forms to include reentry is, as we discussed, to allow forms to attain infinite depth. In the Σ-algebra context this extension to infinite forms is immediate. Make CT_Σ into the initial algebra of $\omega\mathcal{alg}_\Sigma$ by labeling its trees in the obvious manner and adding a $\perp$. Similarly, consider now the set $\boldsymbol{B}_\infty$ of trees of any depth, perhaps infinite but countable, and add to $\boldsymbol{B}_\infty$ an undefined form $\perp_B$ such that $\text{ind}(\perp) = \perp_B$.

Make now $\boldsymbol{B}_\infty$ into an $\omega\mathcal{alg}_\Sigma$ in the obvious manner. Thus $\boldsymbol{B}_\infty$ has an ω-chain-complete carrier, its operations are ω-chain-continuous (since those of CT_Σ are), and there is a unique homomorphism $\text{ind}: \text{CT}_\Sigma \to \boldsymbol{B}_\infty$ that interprets infinite trees as infinite forms. Now we can apply to $\boldsymbol{B}_\infty$ all the results that we have for CT_Σ in general, but that are of interest for reentrant forms in B_∞. In fact, reentry in an indicational form amounts to solving an equation of the form

$$\mathbf{x} = \boldsymbol{\Phi}(\mathbf{x}),$$

where $\boldsymbol{\Phi}$ is any list $\boldsymbol{\Phi} = (\Phi_1, \ldots, \Phi_n)$ of (finite) indicational expressions. We immediately get

Theorem 13.31 *B_∞ is equationally complete.*

Consider now the equational elements in CT_Σ, R_Σ. These are elements of the form $|E|$ such that

$$E: X_i \to F_\Sigma(x_i), \quad 1 \le i \le n, \qquad E(|E|) = |E|.$$

But we know that $|E_B| = |E|_B$. For example, let

$$E(x) = \sigma_1(x) = \overline{x}\urcorner, \qquad E: X_1 \to F_\Sigma(x_1).$$

Then

$$|E| = \bigsqcup_{k \in \omega} \sigma_1{}^k(\perp),$$

and we can interpret in B_∞:

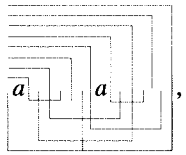
$$\text{ind}(|E|) = |E|_B = \text{ind}\Big(\bigsqcup_{k \in \omega} \sigma_1{}^k(\perp)\Big) = \overline{\overline{\overline{\cdots}\urcorner}\urcorner}\urcorner = |E_B|,$$

satisfying $x = \overline{x}\urcorner$. Quite in general, the infinite solutions in R_Σ, when interpreted, give rise to infinite forms, which are conveniently represented by reentry or reinsertion of a form into itself. Thus

$$x = \overline{x}\rceil = \sqcup$$

is the compact form of the solution $|E|$.

It is important to note that reentrant forms in $\mathrm{ind}(R_\Sigma)$ can arise from more than one equation through mutual interdependence of variables. One such complex reentrant expression is Spencer-Brown's "modulator"

$f =$  ,

which satisfies the following set of equations (Kauffman, 1977) where a is a constant value:

$$x_1 = \overline{ax_2}\rceil,$$

$$x_2 = \overline{ax_1}\rceil,$$

$$x_3 = \overline{x_1x_2x_4}\rceil,$$

$$x_4 = \overline{x_2x_3}\rceil.$$

The solution to this set of equations is an infinite tree constituted by four interdependent infinite trees; f represents the limit of the four interdependent variables. Since we may look at this limit as an unfolding tree, f can also be interpreted as an oscillation in *time* given by the sequences of expressions which the unfoldment determines. In this case, the reader may want to verify that the period of f will be one-half the period of the constant a.

As a sobering note, the following is worth noting. It seems natural to consider equivalence classes in R_B, introduced by a set of initials such as occultation and transposition, which would make R_B into a Brownian algebra. This question, however, is surprisingly complicated, because we have little idea about how to work with elements in R_Σ in general, and with R_B in particular. As a result of this lack of knowledge, we cannot have an idea of, for example, how many arithmetical values are available in R_B. Is it just four, as in V? [For further discussion on this current research, the reader should see ADJ (1978) and Courcelle (1978).]

13.10 Double Binds as Eigenbehaviors[1]

I wish to repeat once more that the main purpose of the detailed discussion of the notion of eigenbehavior over continuous algebras is to give

[1] These ideas were developed jointly with J. A. Goguen. A full account will appear elsewhere.

meat and precision to the invariants that characterize autonomy. These algebraic ideas can be *directly* applied only to the extent that we have a fairly detailed idea of the kinds of operations that are appropriate for some specific systems. The more difficult it is to find precise operational descriptions for the processes present in a system recursion, the more removed that case will be from this particular representation of autonomy.

For example, if we are dealing with the recursions of numerical and logical systems, eigenbehaviors apply directly. In these cases eigenbehavior can be interpreted as the *meaning* or semantics of a process. Consider for example the following recursive process:

$$f(x) = [\text{if } x = 0, \text{ then } 1, \text{ else } x \cdot f(x - 1)].$$

This process is a mixture of Boolean and numerical operators, and, up to the value of x, it defines the factorial function $!_x$. Clearly, however, the recursion involved in the factorial function (process) need not be limited to some fixed x, and in fact, it seems that the meaning of this function should be independent of any specific value of x. In this context, this can clearly be accomplished by taking a function defined by

$$T^i(x) = [\text{if } x \le i, \text{ then } f(x), \text{ else } \bot],$$

and thus we have a chain

$$T^0 \sqsubseteq T \sqsubseteq T^2 \sqsubseteq \cdots$$

with a fixed point

$$! = \bigsqcup_{k \in \omega} T^k(\bot),$$

which is the factorial function. Thus "factoriality" appears as the fixed point of this particular recursion.

To be sure, in the above example we have precise knowledge of the operations involved (Boolean and numerical). When dealing with natural systems the autonomous quality (the semantics or invariance) will be, by necessity, less precise, but also much more rich and interesting. One illustration of this situation is the possibility of gaining a clearer understanding of autonomous units realized in one class of communicational injunctions, the pathological double binds.

The term "double bind" was introduced in the behavioral sciences by Bateson (1959) to describe the mechanism underlying some forms of schizophrenia. The basic insight of this theory was that the etiology of the disease was correlated with some regular pattern of communication within the social matrix. Most frequently this meant communication with the person's family. In a simplified form, this pattern of interaction can be stated as a game following a set of injunctions:

1. start playing the game,
2. produce a certain behavior B,

3. produce the logical opposite of the behavior B (not B),
4. do not leave the game.

A typical instance is that of a child and his mother. By the child's dependency, the first injunction is satisfied. The mother then demands, in an overt verbal form, a behavior such as "love me." Yet in a covert, body-level communication, she rejects the child's response by conveying the message "if you love me you are no good." Again the fourth injunction is fulfilled by the simple inability of the child to exit to another relation. The result is that the child may cut himself off from contact and construct a separate reality. The pathogenic double bind is completed.

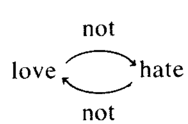
Similar sorts of double bind are very common. The "be spontaneous" variety is perhaps the most familiar. Whenever a behavior is demanded as spontaneous, the very nature of the request makes the demands impossible. Confusion ensues as to what behavior to adopt.

What all such situations have in common is the generation of a *punctuation* of human behavior (Wilden, 1974) in a certain context, that is, the parceling of discrete units of behavior, and the generation of injunctions by communication that operate on these behavioral states in a determined fashion. This can be represented in an operator domain:

$$\Sigma_0 = \text{behavioral states},$$
$$\Sigma_k = \text{injunctions}, \qquad k = 1, \ldots, n.$$

Whenever a social and cultural context has produced such a punctuation, "grammars" of communicative behavior will ensue. In many instances, such behavior will take the form of a finite tree, with exit points into different contexts, or to a different punctuation. Binds arise when trees become infinite, that is to say, when loops arise. Such loops, in our context, can be defined as an eigenbehavior for the equation that defines the infinite tree. The interest in these cases lies in their eigenbehaviors, since they are directly perceived or experienced as undesirable states.

In the double bind mentioned above the states are $\Sigma_0 = \{\text{love, hate}\}$, and operations $\Sigma_1 = \{\text{not}\}$, so that the loop can be represented thus:



In general, in a Bateson-like double bind (2-bind), one has a set of 2 states $\Sigma_0 = \{b_1 b_2\}$, and a tree constituted thus:

$$b_1 \to \text{not } b_1 = b_2 \to \text{not } b_2 = b_1 \to \cdots$$

with the eigenbehavior

$$b^\triangledown = \text{not}(\text{not}(\text{not}(\cdots))) = \text{not}(b^\triangledown).$$

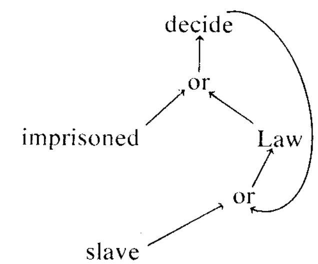
Note that the theory predicts that this eigenbehavior is different from the other states in Σ_0, the initial social punctuation, and hence pathological by the standards of that social context. Such a new state is expressed or has a personal meaning as alienation.

In general then, we define an *n-bind* in human communication as an infinite tree of operations on a set of n behavioral states, whose eigenbehavior is a new state experienced as undesirable.

Negation on 2 states is but one way of producing binds. Consider, for example, the following situation:

1. You must make a decision.
2. Either you are imprisoned or you obey the Law.
3. If you obey the Law, either you live as a slave or you must make a choice.

In this case we have a 3-bind based on the *or* operation, in a loop



A more realistic form of equation would be required to represent one of the many eigenbehaviors described by Laing (1969):

→ it hurts Jack
by the fact
to think
that Jill thinks he is hurting her
by (him) being hurt
to think
that she thinks he is hurting her
by making him feel guilty
at hurting him
by (her) thinking
that he is hurting her
by (his) being hurt
to think
that she thinks he is hurting her
by the fact ┐

Consider here Σ_0 = {Jack, Jill}, Σ_2 = {hurt, think, make guilty} = $\{h, t, g\}$. Then we have the trees:

$$\text{Jill thinks } x \text{ is hurting her} = \begin{array}{c} \text{Jill} \\ \uparrow \\ t \\ \uparrow \\ h \\ \uparrow \\ x \end{array} = t_1;$$

$$\text{it hurts Jack to think } y = \begin{array}{c} \text{Jack} \\ \uparrow \\ h \\ \uparrow \\ t \\ \uparrow \\ y \end{array} = t_2;$$

and

$$t_3 = \begin{array}{c} t_2 \\ \uparrow \\ t_1 \end{array}.$$

The double bind between these two persons arises as the eigenbehavior of the equation

$$x = \begin{array}{c} t_3 \\ \uparrow \\ t_3 \\ \uparrow \\ Jack. \\ \uparrow \\ g \\ \uparrow \\ h \\ \uparrow \\ t_1 \\ \uparrow \\ x \end{array}$$

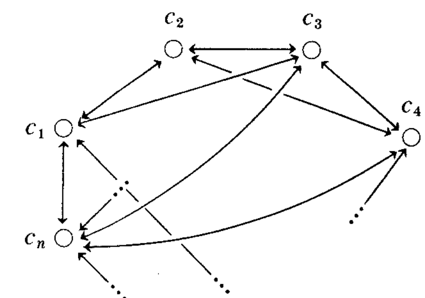
There are, of course, very many questions that we cannot pursue here. For example: What kind of experience would correspond to binds involving more than one equation? How much can the meaning of the states and injunctions (operations) change and still produce the same resulting experience—that is, is there some set of equational constraints valid in human communicational patterns?

13.11 Differentiable Dynamical Systems and Representations of Autonomy

13.11.1

Let us remember that a cellular autopoietic system can be defined as a network of chemical reactions that satisfies two conditions: (1) the chemical species produced are precisely those that constitute the chemical productions producing the chemical species (i.e., closure of the network), and (2) the chemical species produced specify a boundary, physically demarcating the network of productions as a unit in space. The notion of autopoiesis describes the necessary requirements for a class of systems to generate the living phenomenology (Chapters 2-5), but it says little on how to represent this organization. Let us consider now some representations of the cellular case, where closure is accomplished thought chemical transformations.

Imagine a set of chemical species, $c_1, \ldots, c_n$, where there is reciprocal interaction among any subset of them:



(13.4)

The operations acting on the c_i's are production and destruction, and the way to follow what happens is to observe the change in mass of every c_i. Thus let us consider the following operator domain Σ:

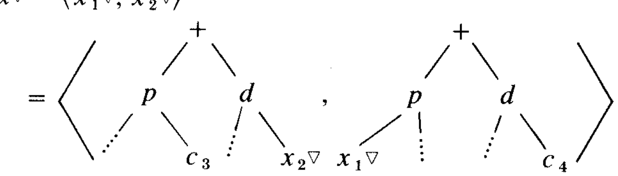
$$\Sigma: \begin{cases} \Sigma_0 = \{c_1, \ldots, c_n\} = \{\text{concentration of } n \text{ chemical species}\} \\ \Sigma_2 = \{p, d, +\} = \{\text{production, destruction, sum of masses}\} \\ \Sigma_k = \varnothing, \quad k \neq 0, 2. \end{cases}$$

We can now consider some specific chemical network, for example,

$$\begin{aligned} x_1 &= p(c_3, x_1) + d(x_1, x_2), \\ x_2 &= p(x_1, x_2) + d(x_2, c_4), \end{aligned} \tag{13.5}$$

with the solution

$$x^\nabla = \langle x_1^\nabla, x_2^\nabla \rangle \tag{13.6}$$



This equation and solution can be rewritten in the more traditional format of chemical reactions:

$$\begin{aligned} c_3 \rightarrow x_1 + x_2 &\rightarrow 2x_2, \\ x_2 &\rightarrow c_4, \end{aligned} \tag{13.7}$$

i.e., x_2 catalyzes its own production. For the eigenbehavior we can write simply

$$c_3 \rightarrow x_1\triangledown \rightarrow \overset{\curvearrowleft}{x_2}\triangledown \rightarrow c_4, \tag{13.8}$$

where c_3, c_4 are given constant concentrations of species c_3, c_4, and $x_1\triangledown$, $x_2\triangledown$ are the concentrations of reactants c_1, c_2.

This network corresponds to one form of the well-known Lotka-Volterra reaction scheme. A simple autopoietic network can be thought of having the general equations

$$\begin{aligned} x_1 &= p(x_1, x_n) + d(x_1, x_2), \\ x_i &= p(x_{i-1}, x_i) + d(x_i, x_{i+1}), \qquad i = 2, \ldots, n, \end{aligned} \tag{13.9}$$

with an eigenbehavior of n interrelated trees $x\triangledown = \langle x_1\triangledown, \ldots, x_n\triangledown \rangle$.

Let us perform the following reinterpretation of this operator domain. First, take the variable x_i as *time*-dependent, real-valued variables. Secondly, interpret the operations p, d as *differential* operators in the variables: production with a positive sign, destruction with a negative one. With this further enrichment of the operator domain Σ, we can rewrite (13.5) in its differential form

$$\begin{aligned} \frac{dx_1}{dt} &= c_4 - kx_1x_2, \\ \frac{dx_2}{dt} &= kx_1x_2 - k'x_2, \end{aligned} \tag{13.10}$$

where constant k, k' represent the rates of the two reactions; that is, we introduce the time dependency in (4),

$$\begin{aligned} c_3 \rightarrow x_1 + x_2 &\xrightarrow{k} 2x_2, \\ x_2 &\xrightarrow{k'} c_4. \end{aligned}$$

Thus in this case, the eigensolution $x\triangledown$ can be related to the differentiable representation by linearizing (13.10) around the steady state and getting the two interrelated solutions—or eigenvalues $\langle \bar{x}_1, \bar{x}_2 \rangle$:

$$\frac{\bar{x}_1}{\bar{x}_2} = \frac{k'}{k'\bar{x}_2}. \tag{13.11}$$

Much study has been devoted to Lotka-Volterra systems of this kind. Although simple, they exhibit a remarkable variety of damped and unstable oscillations depending on the values of the affinities (k, k'), and the perturbations the system is undergoing (fluctuations in c_3, c_4) (see e.g., Glansdorf and Prigogine, 1971).

We have taken this Lotka-Volterra example to this point because it contains, in a nutshell, an important feature and an important limitation of the present approach that should be made clear. As we mentioned in Chapters 7 and 10, the classical notion of *stability* in differentiable dynamics is the only well-understood and accepted way of representing autonomous properties of systems. The work of Thom (1972), Eigen and Schuster (1978), Rössler (1978), Lewis (1977), Bernard-Weil (1976), Rosen (1972) and Goodwin (1976) provides excellent examples of the fertility of this approach for the case of molecular self-organization.

These descriptions look for relevant variables to characterize the coherent, invariant behavior of a unit. Once a set of relevant variables has been identified, a dynamical relation is adopted for the system. This framework for the system's representation has behind it considerable experience from mathematical physics. In the systemic framework, the criterion of distinction for the unit to be studied is given by the invariances resulting from the differentiable description, such as steady states, oscillations, and phase transitions.

An underlying assumption is, however, that there is a collection of interdependent variables, and it is the reciprocal interaction of these component variables that brings about the emergence of an autonomous unit. This is to say that in the instances cited before, the differentiable dynamic description becomes a *specific* case of organizational closure. By adopting the differentiable framework one can mine the richness of the experience behind it. At the same time one finds the limitations imposed by it: More often than not, autonomous systems cannot be represented with differentiable dynamics, since the relevant processes are not amenable to that treatment. This is typical for informational processes of many different kinds, where an algebraic-algorithmic description has proven more adequate.

Accordingly, the fertility of the differentiable representation of autonomy and organizational closure is mostly restricted to the molecular level of self-organization. This is beautifully seen in the work of Eigen and his notion of the hypercycle, recently examined in great detail by Eigen and Schuster (1978); see Figure 4-1. The basic idea here is that a unit of survival in molecular evolution is a closed circuit of reactions with certain structural and dynamic characteristics. Eigen obtains several time invariances for this chemical closure, which serve to illuminate features of the early evolution of life. Also in a differentiable framework, Goodwin (1968, 1976) discusses pathways of metabolic transformation with a view to cellular unity.

13.11.2

There are two comments that are in order at this point. First, it must be noted that Eigen and Goodwin's work is *not* equivalent to a formalization of autopoiesis. This is so because starting from the need to use the differentiable approach, they concentrate on the network of reactions and their temporal invariances, but disregard on purpose the way in which these reactions do or do not constitute a unit in space. Their unit is characterized (is distinguishable) through the time invariances of their dynamics. That is to say, they concentrate on aspect 1 of autopoiesis, but not on aspect 2. This is just as well, for there is much to investigate in just this aspect of recursive chemical networks. It is interesting, however, that the invariances of these systems also reflect space boundaries in some cases, or at least it seems that this could be so in the case of hypercycles. This is considered more explicitly in the well-known ideas of Thom (1972), where a three-dimensional form is associated with a class of dynamics.

A second comment at this point is that a clear distinction should be made between models such as hypercycles, and the analysis of molecular systems through generalized thermodynamics and dissipative structures (Nicolis and Prigogine, 1977). This is so because a dissipative structure takes a complementary view of a unit, namely, it considers the unit as an open, or allopoietic, unit, characterized by the *fluxes* through its boundary. It corresponds to an input–output description in contrast with a recursion description, since the organization of the system takes fluxes explicitly into account in the definition of the environment. In this case, the units distinguished are, strictly speaking, *different* units than the ones distinguish through the closure of some interdependent variables. This is, of course, not to say that there is more merit in one or the other approach. In fact, as discussed in Chapter 10, they have to be viewed as *complementary* characterizations of a system. In the case of dissipative structures, the general allonomous, input–output description is enriched with the differentiable dynamic machinery, and through dynamic variables very detailed results can be obtained. Thus for example, it is possible, in certain cases, to relate explicitly a certain state of flux to the emergence of a spatial boundary, as in the Zhabotinsky reactions.

It is still a matter of investigation how well the differentiable-dynamics approach can accommodate, in a useful way, the spatial *and* the dynamic view of a system. Both on the closure side (e.g., Eigen) or on the input–output side (e.g., Prigogine), there are some striking results showing spatial patterns arising out of recursive, nonlinear reaction schemata. Thus, one can only say that this form of representation has so far provided the most promising approach to coordination, autonomy, and closure at the cellular and molecular level.

But it is in going beyond the molecular level, where we can't rely on a strong physico-chemical background of knowledge, that the insufficiency of the differentiable framework appears, and thus the need to have a more explicit view of the autonomy/control complementarity, and an extension of differentiable descriptions to operational-algebraic ones. A typical borderline case, which we shall examine, is the immune system. A further case where both approaches have been tried is the nervous system. For example, Freeman (1975) prefers a differentiable view that characterizes the time invariances of cell masses, while some [e.g., Arbib (1975)] prefer a more algebraic view, emphasizing cooperation and competition of processes.

We cannot give here an account of how all of these results hold together. In this book I am concerned with emphasizing *one* aspect of systems that has been *neglected*: autonomy. I have offered a characterization of what this means in general, and have provided a representation for some key notions. Thus, for example, autopoiesis, as a case of closure, is not exhausted either in the (possible) algebraic eigenbehavior representation, or in the differentiable-dynamic one. The clear distinction between a class of organizations and its representation must be maintained. Going beyond the differentiable framework of representation was necessary in the past for the allonomy, control viewpoint. Likewise, for the characterization of autonomy, it seems is necessary to go beyond the differentiable framework, while keeping its unique insights for some cases (such molecular organization). The algebraic framework presented above is a step in that direction, though nothing more than a step.

13.11.3

Is there anything useful to say about the relationship between the present algebraic approach to represent autonomy and the classical differentiable one? In some sense one can see that the latter is a specific case of the former, since we deal with some specific collection of operations (differentiation, addition, and multiplication of numerical variables, and so on), and eigenbehavior reduces to the classical notion of stability. This only says to me that the general framework presented here is capable of including this classical picture, and thus lends some credibility to its more encompassing character. However, there are very many detailed questions about the transition from algebraic to differentiable—from eigenbehavior to stability—that are left entirely untouched here, and where more work is needed. Clearly, both approaches cover somewhat *non*-overlapping aspects of systemic descriptions. Thus, it is necessary to have a way of dealing with plasticity and adaptation. Natural systems are under a constant barrage of perturbations, and they will undergo changes in their structure and eigenbehavior as a consequence of them. There is no obvious way of representing this fundamental timedependent feature of system–environment interactions in the present algebraic framework. In contrast, the question of plasticity is a most natural one in differentiable dynamics because of the topological properties underlying this form of representation; hence the notions of homeorhesis and structural stability in all their varieties. To what extent can the experience gained in the differential approach be generalized? How can notions such as self-organization and multilevel coordination be made more explicit in this context? Is category theory a more adequate language to ask these questions? These and many more are open questions.

I offer Table 13.1 to summarize some of the current tools available to present autonomy. In this table I have put the two sides of the autonomy/control complementarity, into correspondence with the two sides of the closure/interactions complementarity. Thus we compare the point of view for characterization of a system with the point of view for its representation. The terms included in the table are simple evocations of notions that are currently more or less well developed. Clearly the lower half of the table is far better developed with regard to mathematical representation. In this book I am concerned with the upper half.

13.11.4

As the reader can well see by now, there is much to say and do in relation to the representations of autonomy. Let me risk being obnoxious in repeating that what I have done here is simply to stake out descriptions which embody, in a mathematical framework, some key ideas that are pursued in this book: closure, autonomy, distinction, recursion. In no way should these formalisms be confused or identified with the intuitions behind them; rather they should be considered only as a vehicle to sharpen precision and reveal inadequacies.

TABLE 13.1.

	Representation	
	Closure	**Interaction**
Characterization		
Autonomy	identity connectivity indefinite recursion eigenbehavior stability	perturbations– compensations cognitive domain resilience ontogenesis
Control	coordination of parts hierarchical levels finite recursion signal flow state transitions	black box dissipative structures input–output

We shall not say anything else about formal representations. Let them stand in their open-ended, incomplete state. In the next and final part of this book, I turn to an altogether different aspect of autonomy, namely, that of the knowledge processes associated with the establishment of a unity. In Table 13.1 it is the upper right-hand corner. In this corner, we look at a unit as autonomous, but in its coupling and interactions with an environment. In this larger view of the autonomous unit, the organizational closure results in a classification of environmental perturbations, and hence in the establishment of a cognitive domain. We now turn to analyze this in detail.

Sources

Goguen, J., R. Thachter, J. Wagner, and J. Wright (ADJ) (1977), Initial algebra semantics and continuous algebras, *J. Assoc. Comp. Mach.* 24:68.

Goguen, J., and F. Varela (1978), Some algebraic foundations of self-referential system processes (submitted for publication).

Varela, F. and J. Goguen (1978), The arithmetics of closure, in *Progress in Cybernetics and Systems Research* (R. Trappl et al., eds.), Vol. III, Hemisphere Publ. Co., Washington; also in *J. Cybernetics 8*:125.