BULLETIN OF
MATHEMATICAL BIOLOGY
VOLUME 39, 1977

A LOGICAL MODEL OF GENETIC ACTIVITIES IN LUKASIEWICZ ALGEBRAS: THE NON-LINEAR THEORY

I. C. BĂIANU, Ph. D., M. Inst. P.,
Faculty of Physics,
Bucharest-Măgurele, P.O. Box 5211,
Romania

A categorical framework for logical models of functional genetic systems is proposed. The logical models of genetic nets are shown to simulate non-linear systems with n-state components and allow for the generalization of previous logical models of neural nets. An algebraic formulation of variable 'next-state functions' is presented which can be used for the description of developmental processes.

1. *Introduction.* One of the first successful applications of Logics to Biology is the use of predicate calculus for a dynamical description of activities in neural nets (McCulloch and Pitts, 1943).

Subsequently, the calculus of predicates was applied by Nicolas Rashevsky (1965) to more general situations in relational biology. Lofgren (1968) introduced a different kind of logical approach to the problem of self-reproduction.

An attempt to provide a characterization of genetic activities in logical terms is presented here. The approach is partially similar to the well known method of McCulloch and Pitts (*loc. cit.*). The main difference arises from the fact that the genes are here considered to act in a step-wise manner. That is, we assume that there exist n discrete levels of intensity of genetic activity.

The "all-or-none" type of activity often considered in connection with genes results as a particular case of our description for $n = 2$.

Our model is a description of genetic activities in terms of n-valued logics.

For operational reasons the model is directly formulated in an algebraic form by means of Lukasiewicz algebras. Lukasiewicz algebras were introduced

by Moisil (1940) as algebraic models of n-valued logics. A further improvement is here made by the use of categorical constructions inside the theory of Lukasiewicz algebras (Georgescu and Vraciu, 1970).

2. *Genetic nets and Lukasiewicz algebras.* Jacob and Monod (1961) have shown that in *E. Coli* the "regulator gene" and three "structural genes" concerned with lactose metabolism lie near one another in the same region of the chromosome. Another special region near one of the structural genes has the capacity of responding to the regulator gene, and it is called the "operator gene". The three structural genes are under the control of the same operator and the entire aggregate of genes represents a functional unit or "operon". The presence of this "clustering" of genes seems to be doubtful in the case of higher organisms.

Rashevsky (1968) has pointed out that the interactions among the genes of an operon are *relationally* analogous to interactions among the neurons of a certain neural net. Thus, it would be natural to term any assembly, or aggregate, of interacting genes as a *genetic net*, without considering the 'clustering' of genes as a necessary condition.

Had the structural genes presented an "all-or-none" type of response to the action of regulatory genes, the neural nets would be dynamically analogous to the corresponding genetic nets. Then, both types of net would be only two distinct realizations of a net which is built up of two-factor elements (Rosen, 1970). This would allow for a detailed dynamical analysis of their action (Rosen, *loc. cit.*, p. 236–249). However, the case which we consider first is the one in which the activity of the genes is *not* necessarily of the "all-or-none" type. Nevertheless, the representation of elements of a net (in our case these are genes, operons, or groups of genes), as black boxes is convenient for formal reasons, and will be maintained in the sequel (see Figure 1).

The genetic net presented in Figure 1 is a discriminating network (Rosen, *loc. cit.*, p. 242). Consider only Figure 1b and apply to it a type of formalization similar to that of McCulloch and Pitts (*loc. cit.*). The level (chemical concentration) of P_1 is zero when the operon A is inactive, and it will take some definite non-zero values on levels '1', '2', . . ., and '$(n-1)$', otherwise. The first level of A is obtained for a threshold value u_0^A of P_2—which corresponds to a certain level of 'j' of B. Similarly, the other corresponding thresholds for levels 1, 2, 3, . . . and '$(n-1)$' are, respectively, u_1^A, u_2^A, . . ., u_{n-1}^A. The thresholds are indicated inside the black boxes, in a sequential order, as shown in Figure 2.

Thus, if A is inactive (that is, on the zero level), then B will be active on the k level which is characterized by a certain concentration of P_2. Symbolically, we write:

$$A(t; 0). \equiv .B(t+\delta; k), \tag{1}$$

where t denotes time, and δ is the 'time lag' or delay after which the inactivity