Chapter 3
Differential Models

Abstract. The models presented capture the intuition of the elementary differential calculus and provide the theoretical substrate for studying designs, schemas, shapes and constructions.

Unconventional frames for time and space, as for instance Galois fields or cyclic groups, have been used to describe the finite or cyclic type of separation and classification processes.

A wave equation is proposed as a differential model for separation and pattern recognition. This model is an abstract complement of transfer equations. The model generates design of experiment matrices as solutions.

Differential posets are introduced as powerful tools in the study of high complexity.

The connection with dual algebras is emphasized.

The notion of a differential category provides a basic axiomatization of differential operators for categories.

3.1 Cyclic Framework

The adopted point of view is that the significance of the process parameters should agree firstly with the mechanism, the nature and goals of analysis for the studied system.

Less-conventional mathematical frameworks are acceptable if these frames can naturally describe the system evolution and the system analysis can proceed on this basis.

Evolvable systems study needs appropriate concepts for time and space (Iordache 2009, 2010, 2011).

Specific chronotopoids should be associated to different reality levels (Poli 2007).

Different kinds of time and of space should be considered in multi-level modelling.

Unconventional models for space of conditions, K, will be presented in the following.

Algebraic finite frames, such as Galois fields and rings or cyclic groups, represent mathematical frameworks that have been used to describe the finite, logical or cyclic type of cognition processes.

A cyclic framework complementing the usual linear one from classical physics proves to be necessary. Evolvability description requires the slow time or cyclic conditions K-processes and the faster, dynamical or linear states S-processes.

Algebraic finite fields represent a common choice for conditions K, whereas the real field is the commonplace structure for states, S. There is a natural hierarchical or cyclic structure associated to finite fields and this explains why they are considered as the appropriate tool for systems structured in conditional levels.

An equation describing the cognitive and self-evolvable systems would contain parameter analogues to the space and the time from the dynamical mathematical models known from physics and engineering.

Consider for example the space Z of vectors describing the properties of an object to be classified and the time T describing the degree of advancement of the pattern recognition, classification, or development, for that object. For the classification process it is possible to associate to different steps in a classification schema digits such as “0” or “1”, with significance as no or yes, true or false, separated or non-separated, identified or non-identified (Iordache et al. 1993a, 1993b, 1993c).

To any object to be classified, a vector Z will be associated in which the properties are specified by digits in the hierarchical order of significance for classification. The same vector Z will give a description of the classification stages in the associated pattern recognition schema. Z describes pattern recognition or stages in problem-solving or development stages for systems and so forth. Denote $Z=z_0 z_1 .... z_j$.

The component z_j should specify the presence of an attribute in classification step, its absence but also partial or uncertain results.

The mathematical framework for Z cannot be limited to that of dyadic that is to Boolean calculus. The need for multi-valued characterization of classification steps and of objects, the uncertainty, imposes mathematical tools completing the better studied dyadic calculations. Detailed description of dynamical systems needs vector characterization corresponding to multi-valued logic such as: “0”, “1”, “2” and so on, meaning, for instance, low, average, high and so on. The coordinate “z_j” characterizes the properties and also it is naturally associated to a stage of classification schemas that make use of the difference in properties noted by “z_j” to perform that kind of classification, pattern recognition, or development.

The degree of advancing in the classification, pattern recognition or in development, denoted by T, was defined as the necessary level of similarity T, between two objects representation, to be classified in the same class (Iordache et al. 1993 a, 1993c). It may be an expansion of the type: $T = t_0 t_1 ... t_j$ with the digits $t_j = 0, 1, 2$ and so on. Denote also this vector by $T= (t_j)$. Each value of T corresponds to another potential step in pattern recognition or in development. Single component vectors with modulo-m algebra structure will be presented as a first example. This is one of the weakest algebraic structures for T and Z still providing a mathematically tractable model adequate to classification and pattern recognition operations or to development study. A slightly different framework to

be considered is that of Galois finite fields. Recall that finite fields with the same number of elements are isomorphic.

Examples of addition and product tables are presented in Table 3.1 and Table 3.2. Here "⊕" denotes the addition and "⊗" denotes the product. The sum and product refers to component-wise operations for vectors as Z or T in K. Here C (m) denotes the modulo-m algebraic framework and GF (m) the Galois field of order m.

C (m) enables to fit the physical intuition concerning the cyclic character of the classification operations in m steps and to justify this first choice for algebraic framework. If m=2 the sum ⊕ is defined as follows: for any two elements T= (t_j) and S= (s_j) the dyadic sum is: $(t \oplus s)_j=((t_j+ s_j) \bmod 2)$. This means that 1⊕1=0, 1⊕0=1. The sum is the dyadic addition, ⊕ equivalent to the dyadic difference. The rule of addition ⊕ signifies that two identical digits have no effects for classification. Only the difference in digits makes a contribution. This addition appears rather as a comparison than as a sum. The product ⊗ is introduced in a way related to cyclic operations too. Product definition takes into account that after m steps the classification process restarts. For the time T or space Z, no change should happen after completion of a cycle of classification operations.

An elementary algebraic framework to be considered is the Galois finite field, GF (m). If m is not a prime number we are faced with rings instead of fields.

Table 3.1 Sum and product in C (m)

C (2)

(x+y) mod2			(x.y) mod2		
⊕	0	1	⊗	0	1
0	0	1	0	0	0
1	1	0	1	0	1

C (3)

(x+y) mod3				(x.y) mod3			
⊕	0	1	2	⊗	0	1	2
0	0	1	2	0	0	0	0
1	1	2	0	1	0	1	2
2	2	0	1	2	0	2	1

C (4)

(x+y)mod4					(x.y)mod4				
⊕	0	1	2	3	⊗	0	1	2	3
0	0	1	2	3	0	0	0	0	0
1	1	2	3	0	1	0	1	2	3
2	2	3	0	1	2	0	2	0	2
3	3	0	1	2	3	0	3	2	1

This algebraic framework was extensively applied in formal logics. For illustration purposes, the operations in GF (3) and GF (4) are presented in Table 3.2.

Let Y denotes the range, the output of a system that performs classification based on features or property examination. Y is element of the same algebraic frames as T or Z. Y may be single-dimensional vector and may assume values 0, 1, 2 and so on, corresponding to various outputs. Multi-dimensional values like $Y=y_0y_1y_2\ldots y_j$ should be examined too. Y, as T or Z, is represented by finite strings. Y definition needs to ensure the logical consistency of the framework.

Appropriate algebraic structures for the range of Y are algebras or fields such as the field of real numbers, the modulo-m algebras, or the finite Galois field, GF (m) that provides physically significant and mathematically tractable models.

Table 3.2 Sum and product in GF (m)

GF (2)

(x⊕y)			(x⊗y)		
⊕	0	1	⊗	0	1
0	0	1	0	0	0
1	1	0	1	0	1

GF (3)

(x⊕y)				(x⊗y)			
⊕	0	1	2	⊗	0	1	2
0	0	1	2	0	0	0	0
1	1	2	0	1	0	1	2
2	2	0	1	2	0	2	1

GF (4)

(x⊕y)					(x⊗y)				
⊕	0	1	2	3	⊗	0	1	2	3
0	0	1	2	3	0	0	0	0	0
1	1	0	3	2	1	0	1	2	3
2	2	3	0	1	2	0	2	3	1
3	3	2	1	0	3	0	3	1	2

The dyadic differential calculus was initially developed for GF (2) situations (Harmuth 1977). If m is a prime-p, the range Y is the standard framework for multi-valued logic. Single-dimensional vectors T, Z, Y are useful if the classification process is based on a single property. For multiple-level cases the parameters T, Z and the functions as Y will be tensor products of single-level cyclic groups.

3.2 First Order Wave Equation

The basic material balance for chemical engineering transport processes is represented by the first-order wave equation:

$$\frac{\partial y}{\partial t} + v\frac{\partial y}{\partial z} + q(y) = 0 \tag{3.1}$$

Here y (t, z) may denote a concentration variable in t-time and z-space, v denotes the velocity, and q(y) denotes a separation rate (Rhee et al. 1989).

The model (3.1) describes many phenomena of physical and technological interest, as momentum, heat and mass transfer.

For mass transfer, the basic model (3.1) shows that the variation of concentration in time is due to the convective process with velocity v and to the kinetic process of interaction, q(y). All the parameters of the model (3.1) are defined on the mathematical real field.

For physical systems study, the real field plays the dominant role.

However this field may be unsuitable to describe some features of high complexity systems.

Paralleling (3.1), a differential model for the space of conditions K, results.

We start with a process of classification, where K is the space of classes.

Classification and the judgment of similarity are fundamental in cognition, serving as the basis for actions. The classification, separation and pattern recognition are the key ingredients in data processing and in problem solving for both natural and artificial evolvable systems. Living or artificially living systems do not survive in environments that they do not recognize or misclassify. Living supposes identification, classification or categorization, separation or combination.

Preliminary attempts for classification or pattern recognition modeling by differential equations outlined the major role of orthogonal arrays (Iordache 2009). A significant result was that the pattern recognition methods parallel screening procedures in experiment design and in problem solving. In particular cases one obtained as solutions of the first-order wave equation orthogonal arrays matrices, Walsh-Hadamard matrices, or Latin squares. Models of cognitive processes such as pattern recognition prove to have as solutions logical thinking methods as that applied in designs of experiments. The result emphasized the deep relation between cognition and evolvability as presented in constructivist perspective and the assertion that both cognition and evolution should be based on similar sets of techniques and models.

To establish the analogs of a dynamical model for classification or pattern recognition, the concept of time and of space in the conditions space K, was adapted to the objectives of the present study.

One of the simplest mechanisms of pattern recognition, classification or development is that in which small changes of the degree of pattern recognition, ∂T, are associated to small changes of the answer, ∂Y. It should be emphasized that the differential is in fact a difference since T and Y are discrete.

Moreover, the change of answer ∂Y depends on both the existing answer Y and the change ∂T of T that is:

$$\frac{\partial Y}{\partial T} \propto Q(Y) \tag{3.2}$$

It is supposed that ∂T is non-null. Otherwise the differential equations are replaced by difference equations. The rate of pattern recognition or classification is denoted by Q. This mechanism is of kinetic type.

Another classification mechanism takes into account that the variation of the answer Y, along the degree of recognition T, is proportional to the answer variation along the features space Z. Classification, pattern recognition and development mean in fact travel in time T, along the space of properties, Z. As Z is screened with a velocity V, the degree of pattern recognition varies proportionally.

This means that:

$$\frac{\partial Y}{\partial T} \propto V \otimes \frac{\partial Y}{\partial Z} \tag{3.3}$$

Here the velocity is a vector $V = v_0 v_1 v_2 \ldots v_j$ or $V = (v_j)$. This mechanism is of convection or drift type.

The general model of the pattern recognition process including both types of recognition processes, corresponding to the evolution according to T and Z, is the first order wave equation WE:

$$\frac{\partial Y}{\partial T} \oplus V \otimes \frac{\partial Y}{\partial Z} \oplus Q(Y) = 0 \tag{3.4}$$

The initial condition is:

$$Y(Z, 0) = F(Z) \tag{3.5}$$

Obviously V and Q may depend on T and Z.

The fact that the addition is equivalent to the difference suggests that a second-order wave equation does not give new solutions in K, as defined.

The mathematical formalism for modeling conditions K apparently follows that of the real states S as a first-order wave equation, WE, but with different addition and product operations taking into account the corresponding domain. Symbolic models, in K, and dynamical models, in S, are complementary in the sense that, neither type of model is reducible to the other. Both are necessary for understanding the whole system including classification schemas and dynamic processes.

The first-order wave equation, WE, is formally similar to the model (3.1) extensively applied in different domains by chemical engineers. For this reason the methodology based on the wave equation, WE, may be considered as a kind of

artificial chemical engineering. It may be related to chemical engineering as the artificial chemistry is related to chemistry (Dittrich et al. 2001) or artificial life to natural life.

The physical, biological or engineering domains offer inspiration for the artificial domains, both for calculus and for artifacts.

3.3 Kinetic Model

For V=0 the first-order wave equation, WE, reduces to the kinetic model:

$$\frac{\partial Y}{\partial T} \oplus Q(Y) = 0 \tag{3.6}$$

The solution in GF (2) is presented here for illustration purposes. In GF (2), "0" denotes the null element. The real product and the sum were translated towards GF (2) operations.

Suppose that the rate of pattern recognition, Q, is the constant expansion denoted by

$Q = q_0 q_1 q_2 \cdots q_j$ or $Q = (q_j)$.

The solution similar to Euler solution for differential equations will be:

$$Y(T) = Y(0) \oplus Q \otimes T \tag{3.7}$$

Recall that the sum $\oplus$ is equivalent to the dyadic difference.

Suppose that, Y (0) = 1. In this case the solution of the first-order wave equation, WE, for different Q is Y (T, Q) and is shown in Table 3.3.

The detailed equations for m=0 are:

$$\frac{\partial y_0}{\partial t_0} \oplus q_0 = 0 \tag{3.8}$$

$$y_0(0) = f_0 \tag{3.9}$$

Table 3.3 Kinetic model, m=0

Q\T	0	1
0	1	1
1	1	0

Table 3.4 Kinetic model, modified, m=0

Q\T	0	1
0	1	1
1	1	-1

Denote, the resulting “0” by”-1”with the same logical signification, for instance “no”. Table 3.4 replaces Table 3.3.

Suppose that Y, T, Q are vectors with two components: $Y=y_0y_1$, $T=t_0t_1$, $Q=q_0q_1$ and $F=f_0f_1$.

This corresponds to two conditioning levels. The first-order wave equation, WE, reduces in fact to two similar equations, one for each level. For m=0 the model is given by equation (3.8) with initial condition (3.9).

For m=0 and m=1 a new equation and initial condition should be added:

$$\frac{\partial y_1}{\partial t_1} \oplus q_1 = 0 \tag{3.10}$$

$$y_1(0) = f_1 \tag{3.11}$$

The fact that one equation in K is replaced by two differential equations, one for each conditioning level, outlines one of the differences between models in K and in S.

Consider the initial condition:

$$Y(Z, 0) = F(Z) = f_0 \times f_1 \tag{3.12}$$

This means that the tensor product is interpreted as a categorical product, denoted by “$\times$”. The selection of tensor product is not limited to this (Iordache 2009).

The solution of the model will be:

$$Y(T) = y_0 \times y_1 \tag{3.13}$$

Specific case $f_0= f_1$, $q_0= q_1$, implies $y_0= y_1$.

Table 3.5 shows the product solution.

Table 3.5 Kinetic model, product $Y(T) = y_0 \times y_1$

1	1	1	1
1	-1	1	-1
1	1	-1	-1
1	-1	-1	1

This represents the Walsh-Hadamard, WH, matrices in DOE. With more coordinates in Y, T, Z it is possible to obtain Walsh matrices with 8, 16, 32, and so on elements.

Table 3.6 Kinetic model, product Y (T) = $y_0 \times y_1 \times y_2$

1	1	1	1	1	1	1	1
1	-1	1	-1	1	-1	1	-1
1	1	-1	-1	1	1	-1	-1
1	-1	-1	1	1	-1	-1	1
1	1	1	1	-1	-1	-1	-1
1	-1	1	-1	-1	1	-1	1
1	1	-1	-1	-1	-1	1	1
1	-1	-1	1	-1	1	1	-1

Table 3.6 shows the three conditioning levels solution for $Y=y_0y_1y_2$ and $T=t_0t_1t_2$ a Walsh–Hadamard DOE matrix.

It was considered that $f_0= f_1= f_2$, $q_0= q_1= q_2$, and this imposes $y_0= y_1= y_2$.

Walsh series as solution of differential equations in dyadic field have been obtained by (Iordache 2009).

3.4 Differential Posets

Differential posets are partially ordered sets of interest for the study of posets, lattices and polytopes. They have been introduced by Stanley in 1988 and since generalized in various different ways (Stanley 1988, Fomin 1994).

A partially ordered set P, or a poset for short, is a pair $P=(S, \leq)$ of a set S together with an order relation $\leq$ which satisfies the following conditions for all x, y, z $\in$ S:

- $x \leq x$ (reflexivity)
- $x \leq y$ and $y \leq x$ imply $x=y$ (anti-symmetry)
- $x \leq y$ and $y \leq z$ imply $x \leq z$ (transitivity)

A lattice is a poset in which any two elements have a unique supremum, the elements' least upper bound, called their join, and an infimum, the greatest lower bound, called their meet. A polytope is a set of vertices in R^n and their convex hull.

Lattice and polytopes play a key role in many modern fields of research, such as algebraic geometry, combinatorics, physics and engineering.

An element x is minimal if there is no y such that $y<x$ and x is maximal if there is no y such that $x<y$.

Given a poset P and elements x, y$\in$ P, we say that y covers x if $x < y$ and there is no z such that $x < y< z$. Then y is a cover of x and x is covered by y and we denote this relationship by $x \prec y$ or $y \succ x$.

For many posets, all order relations follow from the cover relations and transitivity.

A poset with this property is called locally finite.

Every locally finite poset has a naturally associated Hasse diagram. This is a graph whose vertices are elements of the posets and whose edges denotes cover relations, where if $x \prec y$ we draw x below y.

If x and y are elements of the same poset P and $x \leq y$ or $y \leq x$ we say that x and y are comparable.

A chain in a poset P is a set of elements of P which are pairwise comparable while an anti-chain is a set of points which are pairwise incomparable.

A locally finite poset is graded if $P = \cup_n P_n$ is the disjoint union of antichains P_n, indexed by a set of consecutive integers, such that $x \in P_n$ and $y \succ x$ imply $y \in P_{n+1}$ and all minimal elements belong to the same P_i.

Young lattices and Fibonacci-Young are two of the most studied differential posets (Lewis 2007).

Given a non-negative integer n, a partition of n is a finite nonincreasing list of positive integers $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ such that $\lambda_1 + \lambda_2 + \ldots, + \lambda_k = n$.

We denote this by $\lambda \vdash n$. An order on the partitions is defined as follows: given two partitions $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ and $\mu = (\mu_1, \mu_2, \ldots, \mu_j)$, $\lambda \leq \mu$ if and only if $k \leq j$ and $\lambda_i \leq \mu_i$ for $1 \leq i \leq k$. The poset composed of all integer partitions ordered in this way is known as Young's lattice and is denoted by Y.

Young's lattices may be represented as a graph whose vertices are partitions and two vertices λ and μ are adjacent if and only if $\lambda \leq \mu$ or $\mu \leq \lambda$ in the partial order of Y.

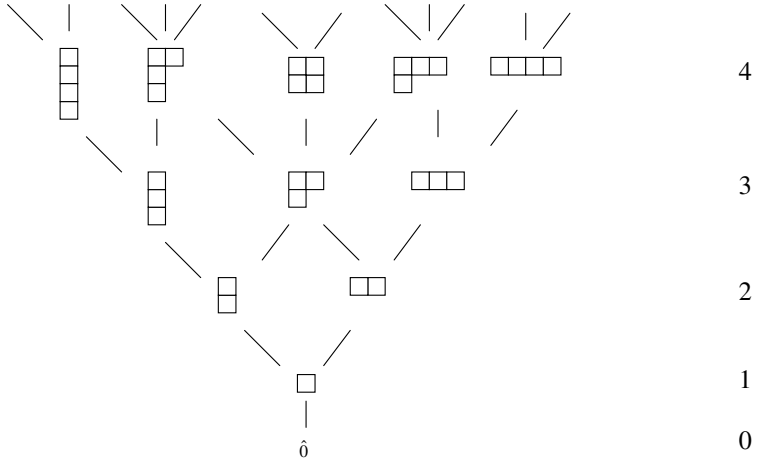

Fig. 3.1 Young lattice

Fig. 3.1 shows a Young's lattice.

Observe that there are five partitions of 4: (4), (3,1), (2,2), (2,1,1) and (1,1,1,1).

A poset P is differential if it satisfies the following three axioms:

DP1. P is locally finite and graded with a unique minimal element denoted by $\hat{0}$.

DP2. If $x \neq y$ are two elements of P and there are k elements of P covered by both x and y, there are exactly k elements of P which cover both x and y.

DP3. If $x \in P$ covers k elements of P then x is covered by exactly k+1 elements of P.

Given any poset P, we may define a vector space: $CP = \oplus_{x \in P} Cx$ of finite linear combinations of elements of P with complex coefficients where no additive relations hold among elements of P.

If in addition P is locally finite and each element of P is a member of only finitely many cover relations we may define two linear transformations U and D on CP as follows:

For $x \in P$,

$$Ux = \sum_{x \prec y} y \quad ; \quad Dx = \sum_{y \prec x} y \tag{3.14}$$

Both U and D should be extended to all of CP by linearity.

The study of the transformations U and D provides information concerning various paths in the Hasse diagrams, the so-called Hasse walks.

Basically Ux keeps track of all possible steps up in the Hasse diagram from x and Dx keeps track of all the steps down in the Hasse diagram.

The behavior of certain combinations of U and D steps on an arbitrary $x \in P$ is of interest for differential posets.

Observe that:

$$UDx = \sum_{y \prec z, y \prec x} z \quad ; \quad DUx = \sum_{z \prec y, x \prec y} z \tag{3.15}$$

It follows that DU-UD=I if and only if P is differential (Stanley 1988).

Moreover for a differential poset P we have:

$$DU^n = nU^{n-1} + U^nD \; ; \; D\hat{0} = 0 \tag{3.16}$$

Thus the action of D on U has a resemblance to that of a differential operator.

This explains the name of differential poset.

Several enumerative results on the class of differential posets have been derived using partial differential equations.

A generalization of differential posets is that of dual graded graphs (Fomin 1994).

A graded graph is a triple G= (P, ρ, E) where:

- P is a discrete set of vertices
- ρ: P→ Z is a rank function
- E is a multiset of arcs (x, y) where $\rho(y) = \rho(x)+1$

The set $P_n=\{x: \rho(x) = n \in Z\}$ are called levels of G.

As for the differential posets the down and up operators can be defined.

Let G= (P, ρ, E) be a graded graph. Linear operators U and D are defined by:

$$Ux = \sum_{(x,y)\in Z} m(x,y)y \quad ; \quad Dy = \sum_{(x,y)\in Z} m(x,y)x \tag{3.17}$$

Here m (x,y) is the multiplicity of the edge (x,y) in E.

Let G_1= (P, ρ, E_1) and G_2= (P, ρ, E_2) be two graded graphs with a common set of vertices and a common rank function.

The oriented graded graph G= (G_1, G_2) = (P, ρ, E_1, E_2) is then the directed graded graph on P with edge in E_1, directed upwards and edges in E_2, directed downwards.

The down and up operators associated with the graph G= (G_1, G_2) are defined by:

$$Ux = \sum_{(x,y)\in E_1} m_1(x,y)y \qquad ; \qquad Dy = \sum_{(x,y)\in E_2} m_2(x,y)x \tag{3.18}$$

Here $m_i(x,y)$ denotes the multiplicity of (x, y) in E_i.

Let (G_1, G_2) be an oriented graded graph such that:

- It has a zero $\hat{0}$
- Each rank has a finite number of elements

Let r be a positive integer.

Then, G_1 and G_2 are said to be r-dual as operators in G= (G_1, G_2) if:

$$DU\text{-}UD=rI \tag{3.19}$$

If G_1 and G_2 are r-dual graphs, we call the pair (G_1, G_2) an r-dual graded graph.

A concept related to derivative poset is that of derivative complexes (Babson and Chan 2000).

Fig. 3.2 shows examples of derivative complexes.

If K is cubical poset, we can define a new cubical poset NK, with elements the ordered pairs (b, c) ∈ K x K such that the join of b and c covers both, while b and c have no meet.

Thus b and c are opposite facets of their join. The partial order on NK is the partial order on K taken component-wise.

Let ε: NK → NK denote the involution ε (b, c) = (c, b)

Then the derivative complex of K is the quotient poset DK=NK/ε.

Observe that NK and DK are both cubical posets and NK is a double cover of DK.

An element {b, c} = (b, c)/ε ∈ DK corresponds to a slice through the interior of the join of b and c, parallel to b and c. An element (b, c) ∈ NK corresponds to the side of {b, c} ∈ DK which faces b in K.

The NK and DK definitions are illustrated in Fig. 3.2.

Recall that a 1-D polytope is a line segment, a 2-D polytope is a polygon and a 3-D polytope is a polyhedron. Higher dimensional polytopes have been studied despite the fact that they are hard to illustrate (Ziegler 1995).

A cubical d-polytope is a d-dimensional convex polytope all of whose boundary faces are combinatorially equivalent to cubes.

The f vector of a cubical complex K is the vector f (K) = (f_0, f_1,…, f_{d-1}) or $\sum_i f_i t^i$, where f_i denotes the number of i-dimensional faces in K.

Babson and Chan proved that:

$$\frac{d}{dt} f(K,t) = f(DK,t) \tag{3.20}$$

Moreover D and N act as derivations with respect to product and disjoint union, that is:

$$D(K_1 \times K_2) = (DK_1 \times K_2) \cup (K_1 \times DK_2) \tag{3.21}$$

Thus the action of D has a resemblance to that of a derivative and this explains the name of derivative complex.

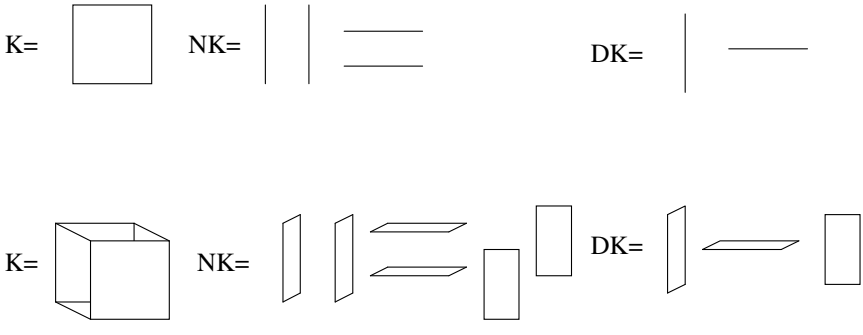

Fig. 3.2 Examples of derivative complexes

Examples of successful collaboration of the posets, lattices, polytopes and differential equations are due to Buchstaber (Buchstaber 2008). The approach is based on a differential ring of combinatorial polytopes. This allows applying the theory of differential equations to the study of polytopes. As an application, Buchstaber described explicitly the generating functions of important families of graph associahedra.

3.5 Doubling and Contracting

The operators U and D keeping track of the up or down steps for differential posets modify just one cell of a Hasse diagram.

A generalization of the differential poset concept is to consider the doubling and the contraction of intervals of lattices diagrams.

Constructive properties of lattice are of interest for the study of lattice boundedness.

A lattice is bounded if it can be constructed, starting with the one-element lattice, by applying a finite sequence of a simple operation called interval doubling (Day 1970).

This operation assigns to a poset P and an interval I, a new poset $P' = P[I]$ by doubling in P the interval I, that is by replacing I in P with the direct product by a two-element lattice.

Denote by "+" the disjoint set union.

The interval doubling construction is defined as follows (Caspard et al. 2004):

Let $(P, \leq)$ be a poset and $I \subseteq P$ an interval of P. Denote by $B=(\{0,1\} \leq)$ the two-element lattice where $0 <1$. The poset P' defined on the set (P-I) + (I x B) is denoted $P' = P[I]$ and is given by the following order: $x' \leq y'$ if and only if:

- $x', \in$ P-I and $x' \leq y'$ or
- $x' \in$ P-I , $y' = yi \in$ I x B and $x' \leq y'$ or
- $x' = xi \in$ I x B, $y' \in$ P-I and $x' \leq y'$ or
- $x' = xi \in$ I x B, $y' = yj \in$ I x B, $x \leq y$ and $i \leq j$ in B.

A lattice L is bounded if there exist a sequence $B=L_1, \ldots,L_i,\ldots,L_p=L$ of lattices and a sequence $I_1, \ldots,I_i,\ldots,I_{p-1}$ such that Ii is an interval of L_i and $L_{i+1}=L_i[I_i]$ for every $i<p$.

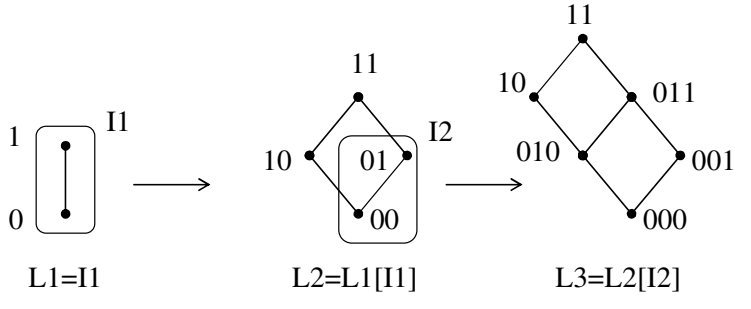

Fig. 3.3 Interval doubling

Fig. 3.3 shows a series of interval doublings starting with the two-element lattice B.

The lattices L1, L2 and L3 are bounded.

Since a bounded lattice is a lattice which can be constructed starting from B by a finite sequence of interval doublings, such a lattice is equally characterized by the fact that it can be contracted until B by an iteration of the operation opposite to the interval doubling. We can call this operation, interval contraction.

First we need to define the gluing conditions:

Let I be an interval of a lattice L, with I equal to the direct product of an interval I0 by B.

We denote by I1, the interval I- I0, isomorphic with I0. We say that I satisfy the gluing conditions if the two following conditions are verified:

- $\forall$ (y,x1,x0) $\in$ (L-I1) x I1 x I0. (y<x1 $\rightarrow$ y$\leq$x0)
- $\forall$ (z,x1,x0) $\in$ (L-I0) x I1 x I0. (z>x0$\rightarrow$z$\geq$x1)

Let L be a lattice and I$\subseteq$ L an interval of L. We say that I is contractible in L if the two following conditions hold:

- I is equal to the direct product of an interval I0 by B. I1 denotes the interval composed of the elements of I-I0.
- The gluing conditions are satisfied on I.

Fig. 3.4 illustrates the interval contracting.

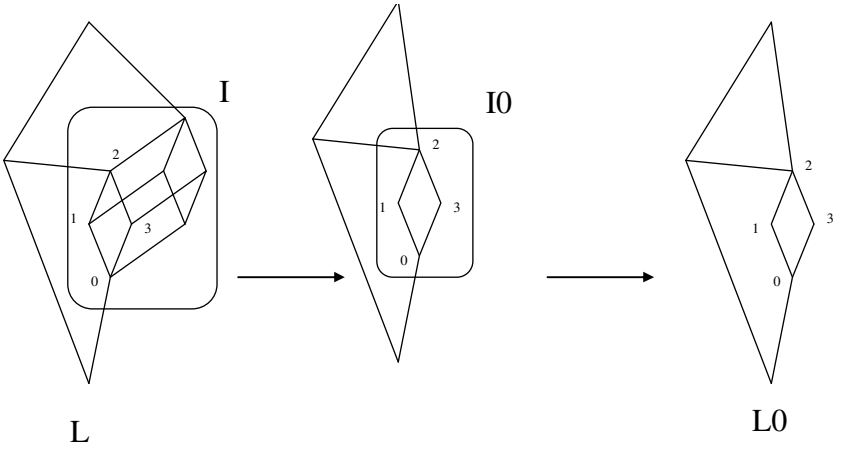

Fig. 3.4 Interval contracting

Let L be a lattice and I $\subseteq$ L a contractible interval of L. We call contraction of I in L the operation of constructing a smaller lattice L0 by replacing I with I0 in L.

The contraction of an interval is the inverse operation to the interval doubling.

Fig. 3.4 shows the contraction of the interval I of the lattice L to the interval I0.

For the lattice L the interval I is contractible to I0.

Doubling and contracting allows obtaining a large variety of lattices generalizing the action of operators U and D defined for differential posets.

Doubling is associated to operator U while contracting corresponds to operator D.

3.6 Hopf Algebras

Important tools for the study of differential posets are the Hopf algebras (Sweedler 1969).

Hopf algebras were introduced, in the context of algebraic topology. The Hopf algebra is an algebra for which the dual space is also algebra so that the duality pairing is intertwined in a specific symmetric way. Examples range from group algebras, their duals, and universal enveloping algebras to deformations of such structures.

More recently the study of these algebras included applications in quantum groups, renormalization and non-commutative geometry.
Hopf algebra has been considered as a unifying framework for modeling several variants of multiplicative linear logic (Blute 1995).

Joni and Rota introduced Hopf algebra in combinatorics (Joni and Rota 1979).

Algebraic structures which are generated by a collection of constructors, like natural numbers generated by zero and a successor or finite lists and trees, are well known in engineering and computer science. Formally they are initial algebras. Induction is used for definition and for proofs for such structures. But there are also important dual coalgebraic structures which do not come equipped with constructor operations but with destructor operations also called observers, transition maps or mutators.

Spaces of infinite data including infinite lists and non-well-founded sets are of this kind. Dynamical systems, with a hidden state space to which the user has only a limited access via specified observer or mutator operations, are coalgebras of different types. Coinduction is the appropriate technique in this coalgebraic context, both as definition and proof principle. The latter involves the so-called bisimulations (Jacobs and Rutten 1997).

Coalgebras are structures that are dual, in the sense of reversing arrows to unital associative algebras. The axioms of unital associative algebras can be formulated in terms of commutative diagrams. Turning all arrows around, one obtains the axioms of coalgebras.

Hopf algebra is a structure that is simultaneously an unital associative algebra and a counital coassociative coalgebra, with these structures compatibility making it a bialgebra (Joni and Rota 1979).

Fig. 3.5 shows the Hopf algebra diagram.

To define the Hopf algebra, H as a k-vector space we need the following elements:

- Product μ: $H \otimes H \rightarrow H$
- Coproduct Δ: $H \rightarrow H \otimes H$
- Unit η: $k \rightarrow H$
- Counit ε: $H \otimes H \rightarrow H$
- Antipode map χ: $H \rightarrow H$

Here H denotes a vector space, k is a discrete field, id denotes the identity operator for H.

With these elements the diagram from Fig. 3.5 will commute.

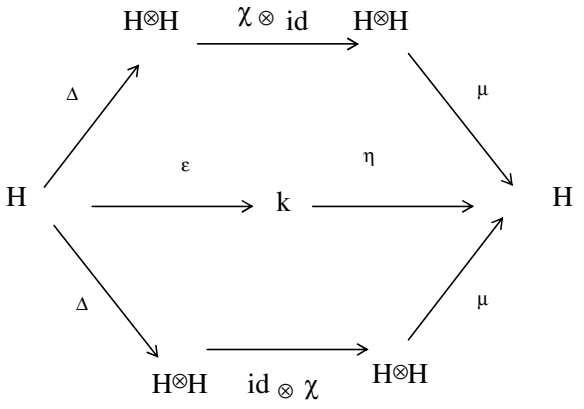

Fig. 3.5 Hopf algebra diagram

We refer to Hopf algebra H with antipode χ.

The coproduct Δ is interpreted as producing sums of pairs of new objects and this explains the potential for combinatorics.

The definition of Hopf algebra is self-dual, as reflected in the symmetry of the diagram from Fig. 3.5. So, if we can define a dual of H, which is always possible if H is finite-dimensional, then this dual is automatically Hopf algebra.

For the self-evolvable systems the concept of dual Hopf algebra is of interest.

Algebra and coalgebra, integration and differentiation are dual concepts.

Fig. 3.6 shows the polytope including algebra and dual algebra.

Fig. 3.6 suggests that after the integration or algebraic way S→K1→K2→K3 we need to look at the differentiation or dual algebraic way K3′→K2′→K1′→S′.

Making use of the developments of the direct way may offer in a kind of symmetry-breaking results. This opens the road for dual Hopf algebras interpretation (Hivert et al. 2005, Nzeutchap 2006). If the two ways offer the same results that is, in the case of self-duality the described system may be evolvable but not self-evolvable.

The swinging from algebra to dual algebra is critical since the boundaries where creative research grows and new information is created consist of parallel tendencies of integration and differentiation. The Self describes the interaction of the two algebras in duality relation.

Swinging method based on dual algebras has been applied in model evaluation and software correcting (Padawitz 2000, Jacob and Rutten 1997).

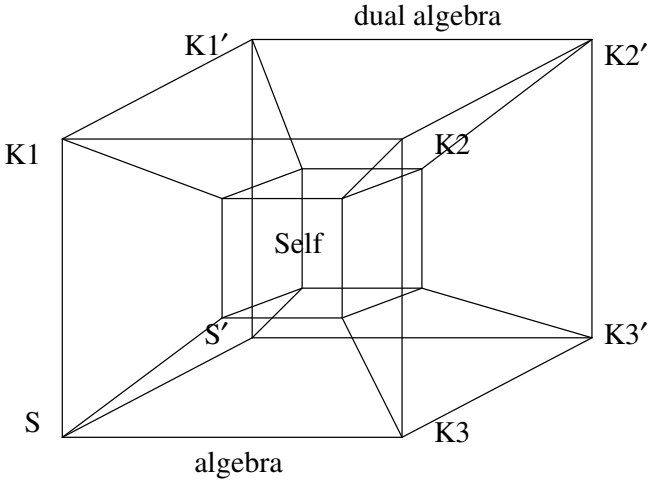

Fig. 3.6 Polytope for algebra and dual algebra

3.7 Differential Categories

Differentiation and differential models have been among the most fundamental tools for scientists and engineers.

Recently several attempts have been made to abstract these notions including approaches based on geometry, category theory and logical intuitions.

It was observed that the obtained new types of models are formally similar to the classical ones. They capture the intuitions from the ordinary calculus since we have calculus rules of differentiation expressed algebraically, formally similar to the classical ones. The new models represent an integral part of what it means to be differentiable.

Attempts to build categories of smooth structures have been suggested and are applied to the general notion of differentiation.

Synthetic differential geometry and linear logic are significant sources of inspiration for differential categories.

According to synthetic differential geometry SDG, an infinitesimal quantity can be taken to be a straight micro segment just long enough to have a slope but too short to bend (Kock 2006). It is an entity possessing location and direction without magnitude, intermediate in nature between a point and a straight line. In a smooth world any interval is indecomposable in the sense that it cannot be split in any way whatsoever into two disjoint nonempty parts.

The SDG provides the conceptual background for development of a mathematically based theory of potentiality and tendency. In conventional approaches, the life trajectory of actual items is characterized by the specific direction that it assumes at any one of its points and by the range of possibilities they have. On the other hand linelets and wavelets considered in SDG are too small to have either probabilities or directions. Instead, they have potentiality and tendency.

The SDG reasoning deals with space forms in terms of their structures that are the basic geometric and conceptual constructions that can be performed on them. The SDG constructions are morphisms which constitute the base category in terms of which we work, the space forms themselves being objects of it. This category is Cartesian closed.

A Cartesian category is a category with a terminal object and for any two objects in the category for instance A, B, the objects and mapping of the categorical product A x B are in the category. The object $\top$ of a category is called a terminal object if for every object in that category there is a unique map from it to $\top$. A category is Cartesian closed if it has all finite products and for any pair of objects say A and B there exists an exponential or map object A^B, the collection of all maps $A \rightarrow B$.

SDG reasoning is based on a category over a natural base topos (Baez 2006). Depending on the nature of the subject under consideration, the corresponding natural geometric form of the objects determines the natural base topos and its logic.

The objects of physics and chemistry have their own geometric form and corresponding logic. If the objects of the theory have a constant and crisp geometric form, we may use classical logic but if the geometric form is variable and fuzzy then we have to use a non-classical more flexible logic, for example the constructive logic.

Another source of inspiration for differential categories is the linear logic (Girard 1987). This is a resource sensitive logic or logic of actions and is based on categorical semantics.

The whole system of linear logic can be considered as an attempt to reconcile the systems of classical logic with the quest for constructive proofs that had led to constructive logic.

Recent developments in the model theory of linear logic have uncovered a variety of models with differential structure.

The development of formal systems for proving computational properties of programs constitutes an important research area of contemporary computer science. Among the aspects needing to be checked, one of the most important is the amount of resources a program will need during its execution. Resources to be bounded can be of different kinds, for instance memory space, or the non-replicable data.

A formal system modeling this idea of resource consumption was presented in Girard's quantitative semantics studies. This semantics establishes an analogy between linearity in the sense of computer science (programs using arguments exactly once) and algebraic linearity (commutation of sums and products with scalars), giving a new mathematically interpretation of resource consumption.

Ehrhard and Regnier designed a formal programming language, called the differential λ-calculus that has a unique kind of argument but two kinds of applications: the usual one and a linear algebra application (Girard 1987, Ehrhard and Regnier 2003). This linear substitution operation can be seen as a formal derivative. Such a syntactic derivative operator can be fruitfully used to increase control over programs executed in environments with bounded resources.

Differential proof nets contain a graph-theoretic syntax for linear logic extended with a differential operator on proofs (Ehrhard and Regnier 2006).

The differential categories and the differential Cartesian categories were defined on the basis of the works on differential λ-calculus (Blute et al. 2006, 2009).

For these categories a derivative operator on morphisms is axiomatized by equations. Typical examples are categories of vector spaces with normed or topological structure allowing defining a notion of smooth maps and of their differentiation.

The notion of a differential category provides a basic axiomatization for differential operators in monoidal categories, which not only generalizes the work of Girard or of Ehrhard and Regnier but also captures the standard elementary models of differential calculus and provides a theoretical substrate for studying a number of non-standard examples.

The basic categorical structure underlying the proof theory of Girard's linear logic (Girard 1987) is a symmetric monoidal closed category equipped with a comonad (Seely 1989).

Monads and comonads are valuable concepts in category theory. To any canonical construction from one type of structures to another, an adjunction between the associated categories, will correspond. Adjoint functors are pairs of functors which stand in a particular relationship with one another. A functor can be left or right adjoint to another functor that maps in the opposite direction. A pair of adjoint functors typically arises from a construction defined by a universal property, and it can be seen as a more abstract and powerful view on universal properties. If F and G represent a pair of adjoint functors, with F left adjoint to G right adjoint, then the composition G◦F will be a monad. The categorical dual of monads, F◦G, will be a comonad. Every adjunction gives rise to a monad. A monad is a functor from a category to itself, in other words an endofunctor. In general, the adjunctions relate categories of different natures. The monad theory tries to capture what is that adjunction preserves. The monads generalize closure operators on posets to arbitrary categories.

The structure necessary to support differentiation is an additive, monoidal category with a coalgebra modality. The morphisms in a differential category should be thought of as linear maps with maps in the category being the smooth maps (Blute et al. 2006).

Denote by $\otimes$ the multiplicative *and*, by 1 the unit, by ! the exponential modalities called: *of course* or *bang*.

A comonad (!, δ, ε) on an additive symmetric monoidal category, is a coalgebra modality in case each object !X comes equipped with a natural coalgebra structure given by:

$$\Delta: !X \rightarrow !X \otimes !X \; ; \; e: !X \rightarrow \top$$

Here $\top$ denotes the tensor unit. This data must satisfy some basic coherence relations (Blute et al. 2006).

The axioms for differential categories are:

D.1 Constant maps:
$D[e_A]=0$

Here e_A denotes the constant map.

D.2 Product rule:
$D[\Delta (f \otimes g)] = (1 \otimes \Delta)\ a_{\otimes}^{-1}(D[f] \otimes [g]+(1 \otimes \Delta) a_{\otimes}^{-1}(c_{\otimes} \otimes 1)\ a_{\otimes}\ (f \otimes D[g])$

Here f: !A→ B, g: !A→ C, and $a_{\otimes}$, $c_{\otimes}$ are the associativity and commutativity isomorphisms.

D.3 Linear maps:
$D[e_A f]= (1 \otimes e_A)\ u_{\otimes} f$

Here f: A→B and $u_{\otimes}$ is the unit isomorphism.

D.4 The chain rule:
$D[\delta\ !f\ g]= (1 \otimes \Delta)\ a_{\otimes}^{-1}(D[f]\ \otimes \delta\ !\ f)\ D[g]$

Here f: !A→ B!, and g: !A→ C!

Observe that according to (D.1) the derivative of a constant is null, and the derivative of a map which is linear is constant according to (D.3). The derivative of the composite of two functions is the derivative of the first function composed with the derivative of the second function at the value produced by the first function as shown by (D.4).

A significant step was to develop an axiomatization which directly characterizes the smooth maps, in other words, to characterize the structure of differential categories directly (Blute et al. 2009). This leads to the notion of a Cartesian differential category. This notion embodies the multi-variable and multi-scale differential calculus.

The basic structure needed for Cartesian differential categories is simpler than is needed for differential categories. It is a left additive category with finite products.

A Cartesian left additive category is a left additive category with products such that the structure maps π_0 π_1 and Δ are additive and that whenever f and g are additive the product is additive.

The axioms for Cartesian differential categories are:

CD.1 $D[f+g] = D[f] + D[g]$; $D[0]=0$
Operators preserve additive structure that is the operator D is linear

CD.2 $D[f]\circ\langle h+k,v\rangle = D[f]\circ \langle h,v\rangle + D[f]\circ \langle k,v\rangle$ and $D[f]\circ \langle 0,v\rangle = 0$
This shows additivity in first argument

CD.3 $D[1]=\pi_0$, $D[\pi_0]=\pi_0\circ\pi_0$ and $D[\pi_1]=\pi_0\circ\pi_1$
Coherence maps are linear differential constant

CD.4 $D[<f,g>]=<D[f], D[g]>$
Operators preserve pairing, that is D behaves coherently with the product structure

CD.5 $D[f\circ g]= D[f] \circ <D[g], g\circ\pi_1>$
This corresponds to the chain rule

CD.6 $D[D[f]] \circ <<g, 0>, <h, k>> =D[f]\circ <g,k>$
Differentials D are linear in first argument.

CD.7 $D [D [f]\circ <<0, h>, <g, k>> = D [D [f]]\circ <<0,g>,<h,k>>$

This means that partial differentials commute.

In Cartesian differential category we obtain partial derivatives from the full ones by zeroing out the components on which the differentiation is not required.

Developments of the notion of differential categories have been proposed (Fiore 2007, Bucciarelli et al. 2010).

Observe that there is a general formalism of differential categories, but there are certain specific ways that, scientists and engineers may use these which turns out to have close analogues in different domains.

References

Babson, E.K., Chan, C.: Counting faces for cubical spheres modulo two. Discrete Math. 212, 169–183 (2000)

Baez, J.: Topos theory in a nutshell (2006), http://math.ucr.edu/home/baez/topos.html

Blute, R.F.: Hopf algebras and linear logic. Mathematical Structures in Computer Science 6(2), 189–217 (1996)

Blute, R., Cockett, J.R.B., Seely, R.A.G.: Differential Categories. Mathematical Structures in Computer Science 16, 1049–1083 (2006)

Blute, R.F., Cockett, J.R.B., Seely, R.A.G.: Cartesian Differential Categories. Theories and Applications of Categories 22, 622–672 (2009)

Bucciarelli, A., Ehrhard, T., Manzonetto, G.: Categorical models for simply typed resource calculi. Electronic Notes in Theoretical Computer Science 265, 213–230 (2010)

Buchstaber, V.M.: Ring of Simple Polytopes and Differential Equations. Proceedings of the Steklov Institute of Mathematics 263, 1–25 (2008)

Caspard, N., Le Conte de Poly-Barbut, C., Morvan, M.: Cayley lattices of finite Coxeter groups are bounded. Adv. in Appl. Math. 33(1), 71–94 (2004)

Day, A.: A simple solution to the word problem for lattices. Canad. Math. Bull. (13), 253–254 (1970)

Dittrich, P., Ziegler, J., Banzhaf, W.: Artificial chemistries-a review. Artificial Life 7(3), 225–275 (2001)

Ehrhard, T., Regnier, L.: The differential lambda-calculus. Theoretical Computer Science 309(1-3), 1–41 (2003)

Ehrhard, T., Regnier, L.: Differential interaction nets. Theoretical Computer Science 364(2), 166–195 (2006)

Fiore, M.P.: Differential Structure in Models of Multiplicative Biadditive Intuitionistic Linear Logic. In: Della Rocca, S.R. (ed.) TLCA 2007. LNCS, vol. 4583, pp. 163–177. Springer, Heidelberg (2007)

Fomin, S.: Duality of graded graphs. J. Algebraic Combin. 3, 357–404 (1994)

Girard, J.-Y.: Linear logic. Theoretical Computer Science 50, 1–102 (1987)

Harmuth, H.F.: Sequency Theory, Foundations and Applications. Academic Press, New York (1997)

Hivert, F., Novelli, J.-C., Thibon, J.-Y.: The algebra of binary search trees. Theoretical Computer Science 339, 129–165 (2005)

Iordache, O.: Evolvable Designs of Experiments Applications for Circuits. J Wiley VCH, Weinheim (2009)

Iordache, O.: Polystochastic Models for Complexity. Springer, Heidelberg (2010)

Iordache, O.: Modeling Multi-Level Systems. Springer, Heidelberg (2011)

Iordache, O., Corriou, J.P., Garrido-Sanchez, L., Fonteix, C., Tondeur, D.: Neural network frames applied for biochemical kinetic diagnosis. Comp. Chem. Engng. 17, 1101–1113 (1993a)

Iordache, O., Valentin, G., Corriou, J.P., Pons, M.N., Pethö, A.: Intermittent Interfacial Transfer. A Dyadic Model. Acta Chemica Hungarica, Models in Chemistry 1(130), 1–18 (1993b)

Iordache, O., Corriou, J.P., Tondeur, D.: Separation Sequencing Use of Information Distance. Canad. Journ. of Chem. Engng. 71, 955–966 (1993c)

Jacobs, B., Rutten, J.: A tutorial on (co)algebras and (co)induction. EATCS Bulletin 62, 222–259 (1997)

Joni, S.A., Rota, G.C.: Coalgebras and Bialgebras in Combinatorics. Stud. Appl. Math. 61, 93–139 (1979)

Kock, A.: Synthetic Differential Geometry, 2nd edn. Cambridge University Press, New York (2006)

Lewis, J.B.: On Differential Posets. Undergraduate Thesis. Harvard College (2007)

Nzeutchap, J.: Dual graded graphs and Fomin's r-correspondences associated to the Hopf algebras of planar binary trees, quasi-symmetric functions and noncommutative symmetric functions. In: International Conference on Formal Power Series and Algebraic Combinatorics, FPSAC, San Diego (2006)

Padawitz, P.: Swinging Types = Functions + Relations + Transition Systems. Theoretical Computer Science 243, 93–165 (2000)

Poli, R.: Three Obstructions: Forms of Causation, Chronotopoids, and Levels of Reality. Axiomathes 17, 1–18 (2007)

Rhee, H.K., Aris, R., Amundson, N.R.: First Order Partial Differential Equations II. Theory and applications of hyperbolic systems of quasilinear equations. Prentice-Hall, Englewood Cliffs (1989)

Seely, R.: Linear logic, *-autonomous categories and cofree coalgebras. Applications of categories in logic and computer science. In: Gray, J., Scedrov, A. (eds.) Contemporary Mathematics, vol. 92, pp. 371–382 (1989)

Sweedler, M.: Hopf Algebras. Benjamin Press (1969)

Stanley, R.: Differential posets. J. Amer. Math. Soc. 1, 919–961 (1988)

Ziegler, G.M.: Lectures on Polytopes. Graduate Texts in Mathematics, vol. 152. Springer, New York (1995)