Universal Eigenvalue Equations

Gabriele Ricci
Università di Parma
I–43100 Parma, Italy.

ORIGINAL VERSION

(This version is from the source the Author typed for Pure Mathematics and Applications, which published G. Ricci, *Universal eigenvalue equations,* Pure Math.and Appl. Ser. B, **3**, 2–3–4 (1992), 231–288.. Unfortunately, the journal retyped it from a printed copy, instead of receiving that source, and did not require proof corrections. The ensuing misprints, particularly the ones for functional notation, are misleading. Also, section numbering is shifted: e.g., present **3.5** corresponds to **4.6** in the published version. In spite of this mishap, the journal was the first to break an eleven years censorship. Before it, only the journal mentioned at the end of the Acknowledgments, tried to provide the Author with informative reports.)

Summary. A part of non-linear mathematics on vector spaces is made universal here, as was done for the linear case by means of Universal Algebra. In order to bridge from linearity to nonlinearity we consider eigenvalue equations (that have an intermediate nature). The main facts, we universalize about such equations, are: the eigenspaces of a matrix form a complete lattice; null eigenvalues characterize singular matrices; eigenvalues and eigenvectors define the general integral of certain difference (and differential) equations; the Cayley–Hamilton theorem holds; each matrix has a (single) geometric dimension and a null determinant implies a matrix singularity.

Contrary to the case of vector spaces, all our treatment is split corresponding to the splitting between the lattice of subalgebras and the one of quotients (that are known to be independent in the universal case). Hence, some of the above facts are peculiar either to the subalgebra case or to the quotient case. (Still, they are not peculiar to vector spaces nor to their numerical fields.)

In connection with characteristic equations, we also universalize the *forms of higher degree* and we study their elementary properties. They concern a *convolution* that universalize the convolution or polynomial multiplication of vector spaces and certain objects and morphisms, related to such forms, with unusual categorical features. Our universal forms are shown to be a "structured abstraction" of Plotkin's quasiendomorphisms. Moreover, when symbolic forms of higher degree replace conventional algebraic terms, some well-known discrepancies between Formal Languages and some programming languages vanish.

We provide the universal definitions with an exhaustive example of "eigenvectors" outside vector spaces. We choose a Boolean algebra and the new objects M, replacing matrices, turn out to be binary clocks. By computing the solutions of all equations we find further minor properties for both the universal case and the Boolean one.

In general, such an M is a family of elements of any algebra, able to identify an algebra endomorphism h, and often it formalizes an object of some computational interest. Since the objects encountered in the applications, as well as the ones undergoing a computation, are the M's instead of the h's, we consider how to handle such "generalized matrices" M. The main tool is a rewording and an extension of the known theory about bases. It merges bases, superassociative systems and a new equivalent notion. A "concrete" characterization of the endomorphism monoid of a free algebra follows from it.

Key words and phrases. eigenspace, innerspace, endomorphism, forms of higher degree, basis, generalized matrices, rheomorphism, general integral, connection algebra, characteristic equation, geometric dimension, universal convolution, shift endomorphism.

AMS Subject classifications: 15A18 and 08A35.

0 Introduction.

0.0 The linearity belief. Here, we study the eigenvalue equations within any algebra. Within a vector space, such equations lead us from linear objects (the matrices) to nonlinear ones (their characteristic equations). This feature will place this topic outside the field of Universal Algebra and this motivates our study.

Hardly, a universal algebraist might seek to universalize any nonlinear mathematics of vector spaces. In fact, whenever he takes a vector space for the algebra of a universal definition or property, he just gets a definition or a property of *linear* mathematics. Furthermore, his universal notions (of linear descent) allow him to handle the nonlinear objects of vector spaces as well, for he gets properties of let us say $f(x) = x^2$ just by jumping from vector spaces to a ring (or even to f itself).

Therefore, a universal algebraist might well believe that *the nonlinear mathematics vanish in the universal case.* Such a belief, indeed, is consistent with the practice in Universal Algebra, where one disregards the original nonlinear meaning of several vector space words, like “algebraic equation” and “polynomial”. Also, it is with some categorical extensions of Universal Algebra, as in [24], where certain monoidal categories, which in a vector space are relevant to linear mathematics, are denoted with the name of “algebraic theories”.

This study of universal eigenvalue equations want to disprove that belief by introducing objects (defined in any algebra) that, within the very vector spaces, reduce to nonlinear objects. Moreover, the monoidal categories do not work there, as we will show in **3.9**, i.e. our universal algebraist will not be allowed to reach those objects by tricks like the jump we did for f.

A motivation for developing new fields of universal mathematics outside Universal Algebra is the following.

So far, only a part of the linear mathematics of vector spaces has been made universal by Universal Algebra: hyperplanes, sheaves, something about linear forms and so on. It is a fundamental part of possible wider universal mathematics, yet — because of this — one hardly should expect many applications. Besides, the vector space applications, corresponding to this small linear part that has successfully been made universal appear negligible relative to the wealth of applications, offered by all of vector space mathematics.

In spite of this, Universal Algebra has found many useful applications that lack any obvious kinship with linear mathematics. Either this is a miracle or the field is full of game. Hence, a further shotgun such as nonlinearity is promising.

Yet, as in vector spaces we pay for the transition from linearity to nonlinearity with more notation and notions, so our notation (in **0.7** and **0.8**) and some notions (in **0.6**) differ from the conventional ones of Universal Algebra, for reasons introduced in **0.4** and proved in the appendix. Therefore, even a reader familiar with Universal Algebra *should not skip this introduction.*

0.1 Eigenvalues outside vector spaces. Let us recall the elementary treatment of the eigenvalue equations of matrices, we want to universalize. When we solve an eigenvalue equation $Mv = \lambda v$ on a complex vector space, we find both any eigenvalue λ and all its corresponding eigenvectors v (which e.g. form a line). Hence, we find any subspace (e.g. of dimension one) of the vector space, where the linear transformation associated to M coincides with the multiplication "times λ", which is any linear transformation of this subspace.

After finding the eigenvectors and eigenvalues of M, we easily get much useful information about M. M is singular iff some λ is, viz. $\lambda = 0$. The v's and λ's allow us to integrate the difference equation $\mathbf{x}_{i+1} = M\mathbf{x}_i$ (or a similar differential equation), given any $\mathbf{x}_0$. An equation of higher degree satisfied by all eigenvalues is satisfied by M also. The minimal degree of such an equation is the geometric dimension of M and, when its lowest degree coefficient (the determinant) is null, M is singular.

The sections from **1** to **4** show that all these notions and properties are not peculiar to vector spaces. We restate and prove everything for any algebra, disregarding any property it might share with vector spaces, up to a minor precaution. We have to split the treatment depending on the two ways an algebra can be part of another: subalgebra or quotient. E.g. in **1.6** the vector space notion of an eigenspace splits into the two different notions of "innerspace" (subalgebra way) and of (universal) "eigenspace" (quotient way). In fact, as known [14, 22], the two ways are highly unrelated in general. (Such a splitting was not needed in vector spaces, where (hyper)planes and sheaves are related by duality.)

The two singularity characterizations of an M in **1.9** use either the innerspaces alone or innerspaces and eigenspaces together. The integration of difference equations in **2.6** uses innerspaces. Hence, the exhaustive computation of of Boolean cospectra, the nine minor proof in **2.1** and the decomposition and integration methods for sequential machines and D0L systems of **2.8** concern innerspaces.

Eigenspaces serve to prove the Cayley–Hamilton theorem in **3.5**, whereas the innerspaces yield a property strictly weaker than Cayley–Hamilton, as shown in **3.6**. Hence, the theory of the "forms of higher degree", developed in section **3** and shown to be non-algebraic in **3.9**, is more relevant to eigenspaces than to innerspaces. The same relevance holds for the proof of the unique geometric dimension of M in **4.1**. Lastly, we prove two determinant properties in **4.2**: one is more relevant to innerspaces, the other to eigenspaces.

If we call preliminary the small part of Universal Algebra that does not involve free algebras, then we might say that here we merely have a preliminary study of universal eigenvalue equations. In fact, we disregard freedom as well as any other special assumption on our algebras.

0.2 "Generalized matrices". The objects, we start with, are families M of elements of an algebra. When this algebra is a vector space, any M is a square matrix, seen as a family of (column) vectors. Hence, our objects are very close to the "generalized matrices" of [15] and the present theory about them could be a part of the theory sought in the open problem 4.6 or **P 1098** ibidem. (These generalized matrices and our families differ mainly because we do not require a base for our algebras and we stay in a single algebra, whereas [15] uses two based algebras.)

When this algebra is not a vector space, our M's are able to precisely model various nonelementary objects, well-known in the applications. In the technical examples or counterexamples, in fact, we shall encounter binary clocks and sequential circuits, D0L and DT0L parallel rewriting systems and directed graphs, while in [34] we can find PERT networks for project planning (i.e. nets of "temporal ands").

Let us see the mechanism underneath this modelling ability. Consider the usual (real three-dimensional) vector space. There, we deal with families $M\colon X \to A$, where $X = \{0,1,2\}$ is the set of versors and A, the carrier of our algebra, is is the set of families from X to the set of reals. Hence, A is the set got from X by a set-theoretical construction (corresponding to our vector space). This works in other cases as well. E.g., when the algebra chosen is the word catenation monoid on an alphabet X, A becomes the set of words on X and M turns out to be the set of productions of a D0L system [36]. I.e. the mere replacement of our set-theoretical construction turns a matrix into another object, without any big loss of theoretical (and practical) properties as seen in **0.1**. (As motivated in **6.12**(A), we consider an object M as a set-theoretical function, possibly satisfying $M\colon X \to A$ for more A's, not as a morphism $\langle X, M, A\rangle$ of the category of sets, in spite of the fact that our set theoretical construction corresponds to a "theory" functor [24].)

As in a vector space a matrix identifies an endomorphism by a representation isomorphism, so in any algebra our M's identify their endomorphisms h. Hence, in many theoretical constructions an h will replace an M. Yet, we stress that our primary objects are the M's. This is motivated by a sort of experimental evidence, coming from all the examples recalled, and is confirmed by a test example, considered in **6.6** for this purpose, concerning partial functions.

All these examples show that the endomorphism ability to model heterogeneous objects is never direct, because endomorphisms always are dull uniform entities (functions from a set into itself), whereas the adaptability comes from their representations M's.

Therefore, we assume that real objects show themselves as such M's and that such M's, or other closely related representations, are the useful ones.

0.3 Analytic theory of bases and independence. There once was an architect who designed a beautiful ellipsoidal vault and wanted to attach some circular lamps at its "most round" points. He asked a friend mathematician for advice. Then, patiently listened the friend (a synthetic geometer) talking about umbilici and so on, waiting for learning where the ladders had to be put under the vault. Unfortunately, when the friend started to chat about "imaginary" circles, he fell asleep.

After choosing concrete objects as in **0.2**, we find ourselves as the architect, who wanted to know where to put the ladders with respect to the walls. We need to know how endomorphisms relate to their representations and how both of them relate to the reference system chosen, in an analytical or "universally cartesian" way, able to direct us toward computations. We also would like to know what happens to our objects when we change the reference system. Unfortunately, present Universal Algebra is unable to handle such "generalized matrices" as our objects are [15]. (So far, the very notions of bases and independence have been treated disregarding any analytic construction, e.g. without any reference to Menger's superassociative systems, even in a treatise such as [23], where such systems are used.)

In section **5** we solve this problem in the case of a single algebra and in the case of independent reference systems, viz. bases, with a fixed index. We reword three known equivalent notions of bases in order to display their underlying analytic object and we add a fourth new definition of a geometric and categorical nature. From the proof of their equivalence we find that these analytic objects are the same object, which is a superassociative system with selectors, up to a minor generalization.

This theorem has several consequences. We can use the singularity criteria outlined in **0.1** to detect independence. (This could reopen the old "solved" problem about breaking independence, recalled in **5.11**.) We can easily change our (generalized) superassociative system depending on a change of the basis. We easily characterize the set of functions, which underlies an endomorphism monoid of a based algebra. (This solves problem 4.7 or **P 1099** of [15] about the "concrete" characterization of free algebra endomorphisms monoids). Moreover, in **5.9**, after a notion of invariance in **5.8** distinguishing the sound properties of algebraic objects from the ones merely dependent on the way we define the objects, we prove that all our properties of universal eigenvalue equations are invariant (whereas a counterexample in **5.10** disproves the invariance of some conditions, which have been a major subject of investigations in Universal Algebra, viz. the "$\mathcal{C}/\mathcal{C}_i$–independences" of [15]).

0.4 Intensionality. Section **1** starts with a rewording of well-known notions of Universal Algebra that has two independent motivations. One, of a foundational nature, will be introduced in **0.6**. Another has a technical nature.

In Universal Algebra (particularly in the categorical one) one often sees objects through their abstract properties, while disregarding many boring set-theoretical construction details. E.g. one identifies a set with its power to a singleton set. Also, functions are introduced only when it is impossible to do otherwise. One says "for all $x \in X$ there is a y such that ..." without mentioning any f such that $y = f(x)$, even though such an f exists by the axiom of choice and even when the y is unique, as it often happens e.g. with categorical limits. The reader's attention has has to be focused on the "..." (a property) rather than on the construction of f. Besides, the sequel can often avoid such hidden f's.

Here, this does not apply any longer. Universal nonlinear objects (as well as the vector space ones) are more complex than "linear" objects and have less predictable properties. Often such a complexity is of a functional nature. Moreover, at the nonlinear level the hidden f's (of the "linear" level) become hard intruders to avoid. Therefore, functions are welcome from the beginning and we build them by functional constructions possibly from other known functions *before* stating their properties (by proofs then made very simple). More in general, we prefer to prove properties of constructed objects rather than to specify objects by properties (and we care for the details of concrete constructions), unless the objects have a mere local use. (E.g., since homomorphisms work at the linear level only, we keep their conventional axiomatic definition, though in **6.9** we show how to make it functional.)

With hidden functions, the relevant property is expressed by the "type" of f i.e. by its domain and codomain. E.g. in **3.1** the universality of direct powers (the only direct products we use here) is expressed by a *bijective type assignment* $f\colon X \mapsto\!\!\!\twoheadrightarrow Y$, where the hidden f is a fundamental functional operator, well-known in Combinatory Logic as the combinator $\mathbf{C}$ (see $\mathbf{C}f$ in **0.8**), and X and Y are sets that serve to express the categorical property of universality set-theoretically. (In this case, we place a construction before a categorical property in order to reach the Cayley–Hamilton theorem quickly.) Another f, useful for the theory of the forms of higher degree, comes from the universality of absolutely free algebras, translated into a type assignment in **1.3**. Such an f turns out to be a basic isomorphism in section **3**.

(We stress that our approach is not a true constructivism We prefer a more conventional foundation such as (classical) Set Theory, disregarding the finer problems of constructivity. Yet, from this nonconstructive ground we move up constructively. We might say that our treatment is intensional relatively to Set Theory. This relative intensionality rises only as a practical compromise, without any claim of logical soundness.

Still it might be useful for (absolute) epistemic intensionality [40] applied to non elementary notions.)

Not all "hidden" functions rise from Universal Algebra. Some will rise and serve strictly within the nonlinear level. Perhaps, hidden functions could serve within Universal Algebra too. In fact, the results relevant to it, just mentioned in **0.3**, follow from a very trivial hidden function, the "CB–function", defined in **1.2**. See **6.12**(B) for more comments.

Anyway, the constructions of such hidden f's always involve few and logically simple "operators" that are scarcely considered in Set Theory. Indeed, they correspond to the combinators of Combinatory Logic [9], [10], [20], while their "types" are akin to the corresponding combinatory types. Therefore, we introduce the combinators into Set Theory (simply as notations extending certain set-theoretical notions).

There are further reasons to use the combinatory notation. Very often, e.g. in the treatment of the forms of higher degree, we have to handle functions with a high "functionality" type. I.e. such functions, after evaluation, provide us with other functions, which, after another evaluation, provide us with other functions and so on several times. This gives rise to several notational problems, which in the case of vector spaces were "solved" for several centuries by ad hoc tricks and special conventions, which are peculiar to vector spaces. (In other fields, e.g. Formal Language Theory, there are different conventions.) Because of our universality purposes, we need a unified notational system for functions. Combinatory Logic provides us a simple and very developed system, similar to other (less developed) systems proposed in [27] or [39] for similar reasons.

0.5 Applications A program solving universal eigenvalue equations has been run on several test examples [34] in order to get a feeling of the present theory. This program accepts certain specifications of finite algebras as (primary) input. Then, it delivers another program able to find the innerspaces of an object M as in **0.2**, assigned in input within the algebra previously specified. The programs, generated in the test runs, turn out to be meaningful in applications. Moreover the user feeds them by the same input, he would use for equivalent hand written programs. This generator shows a heterogeneous output of virtually error free programs, corresponding to the heterogenous nature of our M's.

(This application is an example of a properly universal application, since it exploits the variability of the input algebras. Other applications, not yet verified by experiment, would merely mimic the applications of vector space eigenvalue equations in other (fixed) algebras. One of these improper universal applications is proposed in **6.7**.)

One might see this as a preview of a future "Anumerical Analysis", i.e. of a possible extension of the numerical methods, we presently use for the quantitative features of real life objects, to their non-quantitative features. In fact, once further nonlinear mathematics will be made universal, further "automatic" generators of application oriented programs could also be available. (Possibly, some very high level programming language, e.g. based on algebraic data specifications [12], could organize them.)

This might require more adaptable ways of presenting algebras. The present ways (families of ranked operations) do satisfy the basic algebraic requirements. Yet, a real life algebra needs not to appear as such an extended formalization of a crank adder. Sometimes an unconventional presentation (a few are shown in **6.9**) can be more natural and the reworded algebraic notions of section **1** easily allow it (as shown in [32] as well).

0.6 The notions we assume. "Pure" Set Theory as in [28] is the foundation we follow. It differs from the foundations more used in Algebra and Computer Science (that are "set-theoretical" only in that they use categories directly built from Set Theory). Then, from **1.0** to **1.5** a few Universal Algebra notions have been reworded accordingly. This rewording prevents an inconsistency of conventional treatments, proved in **6.2** and **6.3**. (The mere inconsistency proof appears also in [35]. Here we will also prove that one cannot fix the conventional treatments.)

Also, certain features of real life operations are readily formalized, as shown in **6.0** and **6.1**, and the treatment can be simplified. E.g., once the algebraic terms are really formalized by Set Theory as in **1.2**, algebraic recursion is replaced by a simpler induction and there is a trivially single way to read any term, without any of the "uncoupling lemmata" that many algebraic and categorical works [16] consider fundamental. Yet, as far as the notation is concerned, we have kept the one of Universal Algebra as much as possible. Anyway, this and all other notation not in [28] is listed in **0.7** and **0.8**.

We will (ab)use the word *category* in order to denote a subclass of set-theoretical functions closed under composition and having identities, instead of the corresponding subclass of morphisms of the category of sets. (The set-theoretical functions do not satisfy the last category axiom on morphism disjunction.) Besides, these are the only categories we will consider explicitely.

When using the natural map of a partition, we prefer to introduce it directly. Hence, the usual three-properties definition of a partition is replaced by the *membership property* of its natural map, i.e. we say that a function $f\colon A \twoheadrightarrow B$ has this property when, for all $b \in B$ and all a, $f(a) = b$ iff $a \in b$. It is easy to check that this property is equivalent to the three conventional ones, concerning partition B. We also say that f is a *partition function on (set) A*.

Some Algebra (monoids and lattices) is assumed in the theory. On the contrary, Universal Algebra, Combinatory Logic and some Category Theory are assumed in the introduction, appendix, comments and examples only. Some Computer Science is assumed in some examples.

0.7 Conventional notation. ab denotes the value of function a at b. Accordingly, $r \cdot s$ denotes the composition (in reverse order) of relations s and r. The pair of a and b is denoted by $\langle a, b \rangle$, whereas the usual parentheses denote the syntactic ones only.

Leftmost pairs of parentheses are usually omitted. Infix notation has a lower priority than postfix notation, while postfix notation is lower than the prefix one (including juxtaposition). E.g. $abc \cdot de^{-1}$ means $((ab)c) \cdot ((de)^{-1})$. Sometimes, subscripting replaces juxtaposition for application and the corresponding rightmost pair of parentheses is omitted, e.g. ab_c means $a(bc)$.

The set of functions from A to B is denoted by $\mathsf{F}AB$. (We cannot use the more clear exponential notation, since we often iterate set exponentiations too many times for our interline spaces.) On the contrary, a^n is reserved for the powers of composition. $\mathsf{P}A$ denotes the set of subsets of A. On the contrary, the product of a family A over index J is denoted by $\prod_{j \in J} A_j$ or $\prod_{j \in J} Aj$ and has the lowest priority. The empty set is denoted by $\emptyset$ and the set of natural numbers by N (still we consider $\emptyset = 0$, since we accept the set-theoretical model of N in [28]). An injection, surjection or bijection $f : A \to B$ is denoted by $f\colon A \rightarrowtail B$, $f\colon A \twoheadrightarrow B$ or $f\colon A \rightarrowtail\!\!\!\!\twoheadrightarrow B$ respectively. Given A, by $\leq$ we denote the partial order among the partition functions on A that corresponds to the inclusion among the equivalences on A, i.e. for all partition functions p and q of A, $p \leq q$ iff $q = k \cdot p$ for some function k.

0.8 Combinatory notation and rules. In the theory we use three modifiers and another notation that are set-theoretically defined below and are referred as *combinators*.

- $\mathbf{C}f$ - for $f\colon A \to \mathsf{F}BC$, denotes the function $\mathbf{C}_B f\colon B \to \mathsf{F}AC$, such that $\mathbf{C}_B fba = fab$, for all $a \in A$ and $b \in B$. Our omission of subscript B is a minor set-theoretical abuse, since here B is necessary only when f is empty. When we use this equality to remove a $\mathbf{C}$ from an expression, we say that we perform a $\mathbf{C}$–*reduction*. Clearly, given A, B and C, $\mathbf{C}$ determines a bijection $\mathcal{C}\colon \mathsf{F}A(\mathsf{F}BC) \rightarrowtail\!\!\!\!\twoheadrightarrow \mathsf{F}B(\mathsf{F}AC)$.
  (Notice that, given any $H \subseteq \mathsf{F}BC$, from an $f\colon A \to H$ we still get $\mathbf{C}f\colon B \to \mathsf{F}AC$, because of our foundation in **0.6**.)
- $\mathbf{B}_X e$ - when e is a relation, denotes the relation on families with index X defined componentwise from e, i.e., for all a and b with index X, $\langle a, b \rangle \in \mathbf{B}_X e$ iff $\langle ax, bx \rangle \in e$ for all $x \in X$. Thus, when $e \subseteq A \times B$, $\mathbf{B}_X e \subseteq \mathsf{F}XA \times \mathsf{F}XB$.
  It also denotes the function $\mathbf{B}_X e\colon \mathsf{F}XA \to \mathsf{P}(X \times B)$, A and B being the domain and codomain of relation e, such that for all $a : X \to A$, $\langle x, b \rangle \in \mathbf{B}_X ea$ iff $\langle ax, b \rangle \in e$ for all $x \in X$ and $b \in B$. In the latter case $\mathbf{B}_X ea = e \cdot a$. In both cases, when $f\colon A \to B$, $\mathbf{B}_X f\colon \mathsf{F}XA \to \mathsf{F}XB$ and $\mathbf{B}_X fg = f \cdot g$ for $g\colon X \to A$. A $\mathbf{B}$–*reduction* is the use of any one of these equalities for removing a $\mathbf{B}_X$ from an expression.

(See **6.11** for a motivation of our notation merging abuse and **6.9**(C) for a further combinator **B**.)

$e^{\uparrow}$ - denotes the function $e^{\uparrow}: PA \to PB$, A and B being the domain and codomain of relation e, such that $e^{\uparrow}A' = \{b \mid \langle a, b\rangle \in e \text{ for } a \in A'\}$ for all $A' \subseteq A$.

$\mathbf{i}_A$ - denotes the identity on A.

It is easy to check that our combinators satisfy the following set-theoretical identities that will be widely used in our treatment.

$$\mathbf{B}_X \mathbf{i}_A = \mathbf{i}_{FXA} \quad , \tag{0}$$

$$\mathbf{B}_X(f \cdot g) = \mathbf{B}_X f \cdot \mathbf{B}_X g \quad , \tag{1}$$

$$\mathbf{C}(\mathbf{C}f) = f \quad \text{and} \tag{2}$$

$$\mathbf{C}(f \cdot g)h = \mathbf{C}fh \cdot g \quad , \tag{3}$$

for all applicable functions f, g and h.

Lastly, when we will use an extensionality implication, as "$fx = gx$ for all $x \in X$ implies $f = g$", then we will say that we get $f = g$ by *abstraction of* x, in order to recall the (bound) variable x of the premise.

1 Eigenspaces, innerspaces and singularities.

1.0 Definitions. We will use *operations on* a set A, defined as functions $f: FrA \to A$, where r, the *rank* of f, is *any* set, see **6.1** and [38] for the motivations. In section **6**, f will also be called an Fr*–operation.* A *congruence* of f is an equivalence e over A such that, for all $a, b: r \to A$, $\mathbf{B}_r ea = \mathbf{B}_r eb$ implies that $\langle fa, fb\rangle \in e$. (See **6.11**.) Given $g: FrB \to B$, a *homomorphism from* f *into* g is a function $h: A \to B$, such that

$$h \cdot f = g \cdot \mathbf{B}_r h \ . \tag{4}$$

An *algebra on (carrier)* set A is a family α over Σ of operations $\alpha\sigma: F(S\sigma)A \to A$, where we call the index Σ and the rank family S over Σ its *alphabet* and *species* respectively. (See **6.5** and **6.10**.) Hereinafter, we will deal with a single species S, unless otherwise stated. A *congruence* e of α is a congruence of all $\alpha\sigma$'s.

A *homomorphism from* α *into* β, β being an algebra on B, is a homomorphism h from $\alpha\sigma$ into $\beta\sigma$, for all $\sigma \in \Sigma$. $H\alpha\beta$ denotes the set of all such h's. If h is onto B, then B is the *homomorphic image* of α under h. Such a β is the *quotient* of α by the *partition function* h *of* α, when h has the membership property of **0.6**. ($\leq$ still denotes the partial order of **0.7** on this subset of partition functions.)

If $\alpha = \beta$, we call h an *endomorphism* of α. If $h = \mathbf{i}_A$, we call α a *subalgebra* of β. (Recall that by **0.6** neither h nor $\mathbf{i}_A$ are morphisms of the category of sets.)

1.1 Theorems. *For all algebras* α, (A) *inclusion on the set of congruences forms a complete lattice, where the infimum is given by the intersection on* $\mathsf{P}(A \times A)$*, and similar lattices exist for the quotients or for the partition functions of* α *under* $\leq$*;* (B) *inclusion among the carriers of the subalgebras of* α *does the same (but intersection is on* $\mathsf{P}A$*);* (C) *for all algebras* β *and* γ, $h \in \mathsf{H}\gamma\beta$ *and* $k \in \mathsf{H}\alpha\gamma$ *imply* $h \cdot k \in \mathsf{H}\alpha\beta$, hence composition on $\mathsf{H}\alpha\alpha$ and $\mathbf{i}_A$ form a monoid*;* (D) $h \in \mathsf{H}\alpha\beta$ *iff* h *induces a congruence of* α*;* (E) *if* $C = h^{\uparrow}A$, *for some* $h \in \mathsf{H}\alpha\beta$, *then* $\mathbf{i}_C$ *determines a subalgebra of* β *which is a homomorphic image of* α.

Proofs. (A), (B), (D) and (E) require the same set-theoretical checks of the conventional case. (See **1.2**, **1.4** and **1.8** of [32] for further details.) (C) immediately follows from (0) and (1). *Q.D.E.*

1.2 Definitions. Given any set Y, let $DY = \bigcup(\prod_{\sigma\in\Sigma} \mathsf{F}(S_\sigma)Y)$, i.e. DY denotes the set of pairs $\langle\sigma, t\rangle$ with $t: S_\sigma \to Y$, and, given a set X, consider the minimal set T such that $X \subseteq T$ and

$$DT \subseteq T \ . \tag{5}$$

(We get T by set intersection as §**11** of [28] gets the set-theoretical model of natural numbers and we will go on by mere induction. We will not need a recursive definition, i.e. one that uses X and D to build the recursion operators as in **13.3** or **13.4** of [28].)

If the sets X and DT are disjoint, then we call X a set of *unknowns for* S and T the set of *terms* (symbolic polynomials) *generated by* X *from* S. Under this assumption we call the "pairing" algebra τ on T, defined by $\tau\sigma d = \langle\sigma, d\rangle$ for all $\sigma \in \Sigma$ and $d: S_\sigma \to T$, the *anarchic* algebra *generated by* X. In **6.3** and **6.4** we motivate our assumption that does not allow us to choose any X for a given S. We are also replacing "anarchic", as in [3], for the customary "absolutely free". In case we will need to distinguish our algebras and our anarchic algebras from similar notions as in section **6**, we will call them $\mathsf{F}{\cdot}S$*–algebras* and $\mathsf{F}{\cdot}S$*– anarchic* algebras.

Notice that X, τ and T satisfy three $\mathsf{F}{\cdot}S$*–Peano's* axioms (see [38] for a similar notion):

– *for all* $x \in X$, σ *and* d *as above,* $\tau\sigma d \neq x$,

– $\tau\sigma$ *is one to one and,*

– *for all* $T' \subseteq T$, $T' = T$ *whenever* $X \subseteq T'$ *and* $DT' \subseteq T'$.

This enables us to define functions (or prove predicates) on T by starting with X and by iterations as in (5). We will call this *(*$\mathsf{F}{\cdot}S$*–)term induction* (see **6.3** and **6.4**).

The *(X–anarchic) extension* of an algebra α on A is the algebra $\overline{\alpha}: T \to \mathsf{F}(\mathsf{F}XA)A$ (with a constant species) defined by term induction as

$$\overline{\alpha}xa = ax \quad , \text{ for all } a: X \to A \quad \text{and } x \in X \ , \text{ and} \tag{6}$$

(7) $\overline{\alpha}\langle\sigma,d\rangle a = \alpha\sigma(\mathbf{C}(\overline{\alpha}\cdot d)a)$, for all $\langle\sigma,d\rangle \in T$ and a as above.

Notice that by (3) we can rewrite (7) as

(8) $\overline{\alpha}\langle\sigma,d\rangle a = \alpha\sigma(\mathbf{C}\overline{\alpha}a\cdot d)$.

Each $\overline{\alpha}t\colon \mathsf{F}XA \to A$ is said to be an *algebraic form of* α *over* X. In case we need to recall the set X of generators, T, τ and $\overline{\alpha}$ will be denoted by T_X, τ_X and $\overline{\alpha}^X$ respectively. E.g. $\overline{\tau}_Y^Z$ is the Z–anarchic extension of the anarchic algebra generated by Y.

We will often use a technical notion that does not appear in conventional treatments. Given an algebra β on B, the CB–*function of* α *and* β *at* X is the function $\mathbf{r}\colon \mathsf{F}XA \to \mathsf{F}(\mathsf{H}\alpha\beta)(\mathsf{F}XB)$, defined by $\mathbf{r}gh = h\cdot g$ for all $g\colon X \to A$ and $h \in \mathsf{H}\alpha\beta$, and its value $\mathbf{r}g$ is the CB–*image of* α *and* β *at* g. When, given $M\colon X \to B$, there is a single h such that $M = \mathbf{r}gh$, then h is *the* homomorphism *associated with* M *(with respect to* g*)* and g is a family *associating* h with M, e.g. the family of versors in a vector space associates any linear transformation with its matrix.

1.3 Lemata. *For all* α, β *and* τ *as above,*

(9) $$\mathsf{H}\alpha\beta \subseteq \mathsf{H}\,\overline{\alpha}\overline{\beta} \quad ,$$

(10) $$\mathbf{C}\overline{\alpha} : \mathsf{F}XA \rightarrowtail\!\!\twoheadrightarrow \mathsf{H}\tau\alpha$$

and its inverse is the CB–image $\mathbf{ri}_X$ *of* τ *and* α *at the identity.*

Proofs. In [32] see **1.8** and the proofs of **3.3** and **3.5**. *Q.D.E.*

1.4 Definitions. Let α and τ be as above and let $a\colon X \to A$. The homomorphic image of τ, under the homomorphism $\mathbf{C}\overline{\alpha}a$ associated with a as in (10), and its carrier set $\mathbf{c}a$ as in **1.1**(E) are called the subalgebra of α *connected with* a and the *closure of* a *under* α respectively. Clearly, $\mathbf{c}a$ is the infimum, as in **1.1**(B), of the subalgebra carriers which contain all the ax's. If a subalgebra is connected with some a, then a is called its *generator*. E.g. $\mathbf{i}_X$ *generates* τ. Clearly, any subalgebra has a generator.

1.5 Lemma. *If* α *is connected with* a *as above, then* $\mathbf{r}a$ *is one to one for all* β *as in* **1.2**.

Proof. It is enough to prove that, for all $h, k \in \mathsf{H}\alpha\beta$, $h\cdot a = k\cdot a$ implies $h(\overline{\alpha}ta) = k(\overline{\alpha}ta)$ for all $t \in T$. This follows easily from (9). *Q.D.E.*

1.6 Definitions. In **1.2** take $\beta = \alpha$ and let h be the endomorphism associated with a family $M: X \to A$. Consider a partition function $p: A \twoheadrightarrow Q$ of α and a family $v: Y \to A$, for some Y. Then, consider endomorphisms $m: Q \to Q$, of the quotient γ of α by p, and $n: \mathbf{c}v \to \mathbf{c}v$, of the subalgebra δ connected with v. If, for some m,

$$(11) \qquad p \cdot h = m \cdot p \quad ,$$

then call m an *eigenmorphism of* M (or *for* h) on the *eigenpartition* p. If, for some n,

$$(12) \qquad h \cdot v = n \cdot v \quad ,$$

then call n an *innermorphism of* M (or *for* h) at the *innerfamily* v . To recall M, (11) and (12) can be rewritten as $p \circ M = m \cdot p$ and $M \circ v = n \cdot v$ respectively.

In such cases, γ and δ, as well as their carriers, are called an *eigenspace* and an *innerspace* of M (or h) respectively. (When α is a vector space, δ is the eigenspace under linear algebra terminology.) An innermorphism and its innerspace are said to be *singular* if the innermorphism is not onto the innerspace.

1.7 Lemata. (A) *Given* h *as above, an endomorphism* $m: Q \to Q$ *of a quotient algebra of* α *is an eigenmorphism for* h *iff* m *is a "quotient" of* h, *i.e. iff (11) holds for any partition function* p *of* A. (B) *An endomorphism* n *of a subalgebra* δ *of* α *is an innermorphism for* h *iff*

$$(13) \qquad n \subseteq h \quad .$$

(C) $v: Y \to A$ *is an innerfamily iff* $h \cdot v: Y \to \mathbf{c}v$. (D) *The infimum (in the lattice of the quotients) of a set of eigenspaces is an eigenspace and the intersection of a set of innerspaces is an innerspace.*

Proofs. (A)(Only if) Trivial. (If) Trivial, since Q determines a single p. (B)(Only if) Let $k \subseteq h$ be the restriction of h to $\mathbf{c}v$. Then, $k \in H\delta\alpha$. Therefore, $k = n$ by (12) and **1.5** (use the CB–image $\mathbf{r}v$ of δ and α) and $n \subseteq h$ by construction. (If) Let $v: Y \to A$ be any generator of δ. Then the restriction of h to $\mathbf{c}v$ is n. Hence, (12) is trivial.

(C) (Only if) Trivial. (If) It is enough to check that $k: \mathbf{c}v \to \mathbf{c}v$, k being as in (B, only if). This follows from (9) easily.

(D) Our endomorphism h can be seen as a singleton rank operation h'. Add h' to the other operations of α (by modifying S) and use **1.1** (A) and (B). Then, use present (A) and (B). *Q.D.E.*

1.8 Definitions. The last statement provide us two complete lattices, L' of the eigenspaces and L'' of the innerspaces that we call the *eigenlattice* and the *innerlattice* respectively of M in α. When the zero of L' is the infimum of some set G' of eigenspaces, we say that G' is an *eigengenerator* (of M). Dually, an *innergenerator* is a set G'' of innerspaces, which has the unit of L'' as its supremum. In case every chain in L'' from every $g \in G''$ to the unit of L'' is finite, we say that the innergenerator is *finite chain.*

1.9 Theorems. Given an eigengenerator G' and an innergenerator G'' of M, let E and I denote the corresponding sets of eigenmorphisms and innermorphisms respectively. Then, (A) *the associated endomorphism h is one to one if all eigenmorphisms $m \in E$ are and only if all innermorphisms $n \in I$ are.* (B) *h is onto its domain if all $n \in I$ are and only if all $m \in E$ are.* (C) *When G'' is finite chain, h is singular iff some $n \in I$ is.*

Proofs. (A, if) Let $p: A \twoheadrightarrow Q$ be the eigenpartition for any $m \in E$. For all a', $a'' \in A$, since $m(pa') = m(pa'')$ implies $pa' = pa''$, by (11) $ha' = ha''$ implies $pa' = pa''$. Therefore, since $\bigwedge G'$ is the partition induced by the identity, $ha' = ha''$ implies $a' = a''$.

(A, only if) Trivially from (13).

(B, if) We show that, when no n is singular, h must be onto A. First, note that $B = h^{\uparrow} A$ is an innerspace by **1.7**(B). In fact, by **1.1**(E) B is the carrier of a subalgebra and trivially $h^{\uparrow} B \subseteq B$. Then, we get $A = \bigvee G'' = \bigvee (h^{\uparrow\uparrow} G'') \subseteq B$, since $g = h^{\uparrow} g \subseteq B$ for all $g \in G''$.

(B, only if) Trivial by **1.7**(A).

(C) The "only if" part is in (B, if), since chain finiteness is an additional restriction. Hence, we prove the "if" part (by contradiction).

We will show that, when the endomorphism h is onto A, the function $k = h^{-1\uparrow}$ is able to trace an infinite chain of (singular) innerspaces from any singular innerspace toward A. To get this it is enough to prove that, if B is any singular innerspace, i.e. if it is such that $C \subset B$, where $C = h^{\uparrow} B$, then the set $D = kB$ is a singular innerspace again and $B \subset D$.

D is closed under the operations, since B is and $h \in H\alpha\alpha$. That D is closed under h follows from $h^{\uparrow} D \subseteq B$ and $B \subseteq D$, where the former inclusion comes from the construction of D and the latter comes from $C \subseteq B$ through k, since $B \subseteq kC$. Hence, D is an innerspace.

Moreover, $B \neq D$, because $C \neq B$ and $h^{\uparrow} D = B$, since h is onto A. Hence, we get that both $B \subset D$ and D is singular. *Q.D.E.*

1.10 Comments. (A) It is false that an endomorphism is one to one iff all eigenmorphisms of an eigengenerator are. In fact, take the (one operation) algebra of the sum between natural numbers. Then, the function "double of" is a one to one endomorphism h, whereas in an eigengenerator one might find the eigenspace g corresponding to the congruence mod 2, which preserves h without preserving its injectivity.

(B) In our counterexample, however, the chain of the eigenspaces from g to the zero is infinite. Hence, a finiteness restriction on those chains might still allow us to characterize endomorphism injectivity in the clear cut way we did for ontoness in **1.9**(C). Namely, *if an endomorphism is injective, then all eigenmorphisms within any "finite chain" eigengenerator should be.*

Such a conjecture is relevant to the problem of characterizing Marczewski's independence recalled in 5.11. In fact, it sharpens the (skew) characterization of **1.9**(A).

(C) Statements weaker than our conjecture can be easily proved. E.g., *if an endomorphism h is a bijection, then for any eigenmorphism m either m is a bijection or the eigenspace Q of m leads an infinite chain of eigenspaces toward the zero.*

In this statement, we only need to prove the implication concerning injectivity, since the ontoness implication is trivial, and we do it by a "dual" of the "if" proof of **1.9**(C). Namely, when $h\colon A \rightarrowtail\!\!\twoheadrightarrow A$, we show that, if m is not one to one, then there is another eigenmorphism m', which is not one to one and has a partition function $p'\colon A \twoheadrightarrow Q'$ finer than the one of m, say $p\colon A \twoheadrightarrow Q$.

Define $p' = h^{\uparrow} \cdot p \cdot h^{-1}$. Its membership property immediately follows from the one of p. Then, set $m' = h^{\uparrow} \cdot m \cdot l$, where $l = h^{\uparrow -1} \cdot \mathbf{i}_{Q'}$, and, for all $\sigma \in \Sigma$, set $\gamma'\sigma = h^{\uparrow} \cdot \gamma\sigma \cdot \mathbf{B}_{S\sigma} l$, γ being the quotient of α by p. It easily follows that h and γ have m' and γ' respectively as quotients by p' and that m' is an endomorphism of γ'. Clearly, m' is injective only when m is and, since $p = (l \cdot m') \cdot p'$, $p' < p$ as we claimed.

2 You don't need numbers to do integrations.

2.0 Comment. Let us see how to translate the ad hoc notions of Switching Theory about Boolean set algebras into the universal ones of section **1**. In fact, in **2.1** we choose our M's in such algebras in order to concretely introduce nonlinear problems about nonnumerical objects.

Consider an engineer designing a simple autonomous circuit, i.e. a binary clock. At first, he chooses a number of "states" that be an n–th power of two. On such a set Y of states, he assigns a "binary encoding", viz. each state is identified by an n–tuple of binary digits.

From our point of view, he chooses an associating family $g : X \to A$ as in **1.2**, where $X = n$ and $A = \mathsf{P}Y$ with $Y = \mathsf{P}X$ (see also **I.3.19** of [24]), such that e.g. $gx = \{y \mid \ x \in y\}$ is the set of states "bearing a 1 at the x–th bit". (Recall that $n = \{0, ..., n-1\}$ by our foundation choice of **0.6**.) This always is an associating family for the algebra α of the finitary Boolean operations on the subsets of Y, because of **1.5**, as one can easily check.

Now, our engineer defines its clock by the n Veitch diagrams (Karnaugh maps), drawn from the desired state transition map $\delta\colon Y \to Y$. By the x–th diagram he denotes the set of the states going in a state bearing a 1 at the x–th bit. Then, he realizes the clock by n Boolean polynomials determined by the sets of states denoted by each diagram. (These "polynomials" provide him a logical wiring for the clock hardware.)

From our point of view, he has set a family $M : X \to A$ (as in **1.2** with $\beta = \alpha$) at the diagram values $Mx = \{y \mid x \in \delta y\}$. (More in detail, a Veitch diagram is the characteristic function $mx\colon \mathsf{F}X2 \to 2$ of an Mx. Replacing all sets by their characteristic functions is awkward theoretically, yet useful practically. Notice that $m = \mathbf{C}\delta$.) Therefore, we have an associated endomorphism h of α, $h \cdot g = M$, where our treatment applies. Then, each set of states Mx determines a "conjugate" (Boolean) function, in the same way as a column vector Mx of a matrix M on a vector space A determines a (linear) form. Section **5** will provide us details about such a conjugation in the universal case.

(Notice that the engineer never uses h. All he needs is M, either as an m or as a δ.)

2.1 Example. We are familiar with eigenvectors and eigenvalues in vector spaces. In order to gain such a familiarity elsewhere, we resort to an exhaustive computation of the singleton innerfamilies of all endomorphisms of a finite algebra. In fact, the value of such an innerfamily (an *innerelement*) can be thought of as an "innervector" in a somehow strange vector space.

Boolean set algebras compare well with vector spaces, as far as algebraic properties are concerned, and we choose one of them. Four atoms, i.e. two bits, suffice to provide us a general view. There are 256 endomorphisms in this algebra that correspond to 256 two–bits binary clocks, i.e. doubleton families M or pairs of conjugate functions $x' = p(x, y)$ and $y' = q(x, y)$, p and q being the Boolean realizing polynomials.

We fix the sequence $(0, 1, 2, 3)$ of our four atoms in order to assign any state transition function as a corresponding sequence $(\delta 0, \delta 1, \delta 2, \delta 3)$ (see the sequences in the column "next states" of the following table). The four atoms are encoded by the standard positional two–bits encoding of the first four natural numbers, viz. we choose $g0 = \{2, 3\}$ and $g1 = \{1, 3\}$. Under this encoding, for each clock, we have the sets $M0$ and $M1$ listed in the table column M that correspond to the polynomials p and q listed in the columns x' and y'. Here, we omit commas, parentheses and braces, when we denote sequences or sets. (Thus, $0 \neq \emptyset$ here.)

In order to find all innervectors of a clock, by **1.7**(C) we merely check for each set v of states that $hv \in \mathbf{c}v$. After running this check for a while, we find that:

(A) both algebraic constants of α are (trivial) innervectors,

(B) isomorphic clocks have isomorphic innermorphisms and

(C) all non trivial innerspaces are isomorphic to the Boolean lattice on the atoms 0 and 1 under an isomorphism that preserves the innermorphism,

These first findings suggest us to ignore constant innervectors (as customary in vector spaces), to gather isomorphic clocks by counting the clocks isomorphic to a given one (the <u>n</u>umber of <u>i</u>somorphic <u>c</u>opies is given in the "nic" column) and to denote any innermorphism by its "innervalue", viz. by one of the four elements of the lattice in (C) which (as singleton family value) associates the innermorphism.

We choose atom 1 as the (value of the singleton) associating family for the latter Boolean algebra and we get the four innervalues in the heading of column "Cospectrum".

For each of them and for each clock we score the total number of innervectors (neither a multiplicity number nor the number of innerspaces). The innervectors are shown in their column by lists, separated by semicolons, corresponding to each entry of the cospectrum column. Each list bears innervectors, separated by commas, or the symbols $\cup$, denoting all innervectors got by unions from innervectors in the list, and $\cap$, doing the same but by intersections.

Next	*M*		x'	y'		*Cospectrum*				*Innervectors*
states	$M0$	$M1$			nic	$\emptyset$	1	0	01	
0 1 2 3	23	13	x	y	1	–	14	–	–	*all*
0 0 0 0	$\emptyset$	$\emptyset$	0	0	4	7	–	–	7	$1, 2, 3, \cup; \overline{1}, \overline{2}, \overline{3}, \cap$
0 1 2 2	23	1	x	$\overline{x}y$	12	1	6	–	1	$3; 0, 1, 23, \cup; \overline{3}$
0 1 1 1	$\emptyset$	$\overline{0}$	0	$x+y$	12	3	2	–	3	$2, 3, \cup; 0, \overline{0}; \overline{2}, \overline{3}, \cap$
1 0 0 0	$\emptyset$	0	0	$\overline{x}\,\overline{y}$	12	3	–	2	3	*as above*
0 0 2 2	23	$\emptyset$	x	0	12	3	2	–	3	$1, 3, \cup; 01, 23; \overline{1}, \overline{3}, \cap$
2 2 0 0	01	$\emptyset$	$\overline{x}$	0	12	3	–	2	3	*as above*
0 2 1 3	13	23	y	x	6	–	6	–	–	$0, 3, 12, \cup$
1 2 2 2	$\overline{0}$	0	$x+y$	$\overline{x}\,\overline{y}$	24	3	–	–	3	$0, 3, \cup; \overline{0}, \overline{3}, \cap$
0 0 1 1	$\emptyset$	23	0	x	12	3	–	–	3	$2, 3, \cup; \overline{2}, \overline{3}, \cap$
1 0 3 2	23	02	x	$\overline{y}$	3	–	2	4	–	$01, 23; 02, 13, 03, 12$
1 0 2 2	23	0	x	$\overline{x}\,\overline{y}$	12	1	2	–	1	$3; 01, 23; \overline{3}$
3 1 2 2	$\overline{1}$	01	$x+\overline{y}$	$\overline{x}$	24	1	2	–	1	$0; 1, \overline{1}; \overline{0}$
2 1 0 0	0	1	$\overline{x}\,\overline{y}$	$\overline{x}y$	24	1	2	–	1	$3; 1, \overline{1}; \overline{3}$
2 3 0 0	01	1	$\overline{x}$	$\overline{x}y$	24	1	–	2	1	$1; 01, 23; \overline{1}$
0 2 3 1	12	23	$x \oplus y$	x	8	–	2	–	–	$0; \overline{0}$
1 3 2 2	$\overline{0}$	01	$x+y$	$\overline{x}$	24	1	–	–	1	$0; \overline{0}$
1 1 0 0	1	0	$\overline{x}y$	$\overline{x}\,\overline{y}$	24	1	–	–	1	$3; \overline{3}$
2 0 3 1	02	23	$\overline{y}$	x	6	–	–	2	–	$03, 12$

The rows of the table have been ordered from the delay and reset machines to the four states counter by an increasing (visually intuitive) complexity of the state diagrams, as the reader might check by drawing them. From this shrunk table we easily get the following findings.

(D) Some "cospectrality" appears among clocks that are not onto (see the middle of the table), viz. there are nonisomorphic clocks, with an endomorphism or a state table that is not onto its domain, which share the same cospectrum, whereas

(E) no cospectrality appears among onto clocks.

(F) The lists of the innervectors corresponding to innervalue 1 are closed in α, whereas in general

(G) the sets of innervectors sharing the same innervalue are not always closed in α, e.g. the second list for the clock "1 0 3 2" does not.

(H) If an innervector v has an algebraically constant innervalue c, then $\overline{v}$ is an innervector with value $\overline{c}$.

(I) There are no empty cospectra, and

(J) many cospectra are *real*, viz. many clocks have innervectors enough to build an innergenerator, yet

(K) not all cospectra are real. (See the clocks from "0 2 1 3" to the end, but for "1 0 3 2".)

Some findings are self-proving counterexamples: (D), (G) and (K). A few others are educated guesses: (E) and (I). Many others are easy theorems for Boolean set algebras, as (C) and (H), or for universal ones up to minor rewordings, as (A), (B) and (F). ((C) follows from the innerspaces being generated by a single nontrivial innerelement, (H) from the preservation of the complement by the innermorphism, (A) from algebraic constants being fixed points of endomorphisms, (B) from (13) and (F) from the innermorphism being the identity and from **1.7**(C).)

2.2 Comment. Consider a system of homogeneous difference equations of the first order with constant coefficients for a real vector space of finite dimension, say $v_{i+1} = h(v_i)$ or $v_{i+1} = M \circ v_i$, h being a vector space endomorphism and M being its matrix. We know how to integrate it by exponentials "on" eigenvectors and trigonometric functions "on" eigenplanes. The set of its integrals is conveniently expressed by a *general integral* and by the set of the *initial conditions* determining each integral.

A further finding from the preceding example is that this method works even for our clocks. You can check it on the real cospectra of **2.1**(J) by the "general integral" $v_i = f(\mathbf{S}n^{(i)}w)$, where i is any natural number, w is a family over some index Y of innervectors ranging over an innergenerator, $n^{(i)}$ is the family with index Y of the i–th composition powers of the corresponding innermorphisms ny, $\mathbf{S}n^{(i)}w$ denotes the family with index Y of the sets of states $(ny)^i(wy)$ and f is any Boolean algebraic form over Y that works as the *parameter* of the general integral. (A reader, wishing to check it now, might choose the "unbiased" clocks "0 1 2 2", "2 2 0 0" and "1 0 3 2". Anyway, the last one will be considered in **2.8**.)

Before we state and prove such a finding for any algebra, notice that the cases considered here correspond to the integration by exponentials only. By **2.1**(K) we need to consider the "trigonometric" case too, i.e. to allow a general integral to be "on" eigenspaces other than the singleton generated ones. This will entail some more formalism.

2.3 Definitions. Let α and h be as in **1.6**. Given a family Y with index Z, U denotes its disjoint union $\bigcup(\prod_{z\in Z} Y_z)$ and, given a family w with index Z of families w_z with indexes Y_z, uw denotes the family with index U such that $uw\langle z, y\rangle = wzy$ for all $\langle z, y\rangle \in U$. Also, g denotes any algebraic form of α over U. Thus, for all $a : U \to A$,

(14) $$h(ga) = g(h \cdot a) \ ,$$

as it follows from (9). We call a family w with index Z of innerfamilies $wz: Y_z \to A$ an *innergenerator family* if it spans A, viz.

(15) $$\mathbf{c}(uw) = A\ .$$

Consider the recursive equation

(16) $$v_{i+1} = hv_i$$

on the unknown sequence $v: N \to A$. We call *general integral* of (16) a family $\mathcal{S}: G \to \mathsf{F}NA$, where index G is called the set of *parameters*, of sequences that are all and only its *integrals*, i.e. all and only the v's that satisfy (16).

2.4 Lemma. *A family $\mathcal{S}: G \to \mathsf{F}NA$ of integrals is a general integral iff $\mathbf{C}\mathcal{S}0$ is onto A.*

Proof. Trivial application of the iteration principle, e.g. see **13.2** in [28]. *Q.D.E.*

2.5 Definition. Let $w \in \prod_{z\in Z} \mathsf{F}Y_zA$ for some Z and Y, n be a family with index Z of endomorphisms n_z of the subalgebras on $\mathbf{c}w_z$ and $f: N \to \mathsf{F}UA$ denote the function such that, for all $i \in N$, $z \in Z$ and $y \in Y_z$,

(17) $$f_i\langle z, y\rangle = n_z^i(wzy)\ .$$

The family $s: G \to \mathsf{F}NA$, where G is the set of algebraic forms of α over U, such that

(18) $$sg = g \cdot f$$

for all $g \in G$, is called the *w–n form* of h.

2.6 Theorem. *A w–n form is a general integral iff w is an innergenerator family and n is its family of innermorphisms.*

Proof. (If) We first prove that each sg is an integral. From (17) and (13) we get $h(f_i\langle z, y\rangle) = f_{i+1}\langle z, y\rangle$. Therefore, since by (18) and (14) $h(sgi) = g(h \cdot f_i)$, we have that $h(sgi) = sg(i + 1)$ by abstraction of $\langle z, y\rangle$.

Conversely, given any integral v with $v_0 = a$, we resort to **2.4** to prove that $v = sg$ for some algebraic form g. In fact, by (15) we always can find a g such that $sg0 = g(uw)$ is any a.

(Only if) We first prove we have innerfamilies and innermorphisms. For each $z \in Z$ let $g': Y_z \to \mathsf{F}(\mathsf{F}UA)A$ be the family of projections $g'y = \mathbf{Ci}_{\mathsf{F}UA}\langle z, y\rangle$ with $y \in Y_z$. Since $d = s \cdot g'$ is a family of integrals, $h \cdot \mathbf{C}d0 = \mathbf{C}d1$. This easily yields the innervalue equations $h \cdot w_z = n_z \cdot w_z$. (Apply y to both sides, use (18) and (17) through $\mathbf{C}$–reduction, then abstract y.)

Now, let us check (15). Since by (17) $\mathbf{C}s0 = \mathbf{Ci}_G(uw)$, by **2.4** there is $g \in G$ such that $g(uw) = a$ for each $a \in A$. *Q.D.E.*

2.7 Comments. (A) Let us consider how to replace the general term induction of **1.2** for the induction on natural numbers used in the previous theorem (where the numbers were the exponents of composition powers of a single endomorphism). In (B) and in **3.9**, this will allow us to extend some results, by replacing certain terms for natural numbers.

The induction on natural numbers is a case of the term induction of **1.2**. In fact, the set of terms, generated by a singleton set of unknowns from a singleton species with a singleton rank, is a model of N. Hence, we have to consider how to remove the singleton restrictions on the latter species, which we call *primary*.

We can introduce any primary alphabet I, once we have a family of endomorphisms $h{:}\, I \to \mathsf{H}\alpha\alpha$, as it happens e.g. for the next applications in **2.8**(B) and (C). In fact, the letter of the singleton species served to count how many times an endomorphism is composed in a composition power to a natural number. More primary letters just enable us to compose more endomorphisms. On the contrary, it is harder to remove the singleton restriction on the ranks, since a primary rank has to index the single argument of an endomorphism. Hence, we also can safely keep the singleton restriction on the set of unknowns and we take it as the one for natural numbers, $\{0\} = \{\emptyset\}$.

Therefore, instead of natural numbers we can use terms that are singleton generated from any species as above, viz. from any *unary* species. We simplify these terms by simplifying (5) according to our singleton ranks, viz. we inductively define their set I^*, the set of *words on I*, by $\emptyset \in I^*$ and $I \times I^* \subseteq I^*$. Hence, we replace word induction for natural induction.

We stress that our words come from unary species, not from word catenation monoids. (When such monoids are (mis)used, e.g. in Regular Language theory, ad hoc notions, like as the "right/left homomorphism", have to replace the standard ones.) Terms of other species, i.e. trees, could serve as "exponents" only in wider extensions (together with "homomorphisms with many arguments".)

(B) In the preceding theorem we still had (natural) numbers as integration variables. Yet, numbers are not needed even in this minor role. In order to get rid of them we use other discrete objects, the previous words, as integration variables. We assume a family $h{:}\, I \to \mathsf{H}\alpha\alpha$ instead of a single endomorphism. Then, we extend equation (16) into $v\langle\sigma, t\rangle = h_\sigma(vt)$ for all $\sigma \in I$ and $t \in I^*$, with an unknown family $v{:}\, I^* \to A$. We solve it in the same way, but for few changes.

Define *I–innerfamilies, I–innerspaces* and *I–families of innermorphisms* by modifying **1.6** in an obvious way: an innermorphism for each $\sigma \in I$, but all of them on the same I–innerspace, generated by the same I–innerfamily. Accordingly, we keep the definition in **2.3** of innergenerator family to define an *I–innergenerator family*, whereas its family of I–families of innermorphisms, as well as the family of families of endomorphisms of **2.5**, becomes an $n \in \prod_{z\in Z} \mathsf{F}I(\mathsf{F}(\mathbf{c}w_z)(\mathbf{c}w_z))$. Thus, (17) becomes $f_t\langle z, y\rangle = (\underset{t}{\bigodot}_\sigma nz\sigma)(wzy)$

for all $t \in I^*$, $z \in Z$ and $y \in Y_z$, where $\bigodot$ is a *composition product*, i.e. $\bigodot_{\emptyset}{}_{\sigma}\, nz\sigma = \mathbf{i}_{\mathbf{c}(wz)}$ and $\bigodot_{\langle\varsigma,t\rangle}{}_{\sigma}\, nz\sigma = nz\varsigma \cdot \bigodot_{t}{}_{\sigma}\, nz\sigma$.

Once these extensions are done, the previous theorem follows easily. Just add quantification $\sigma \in I$ where needed.

2.8 Examples. (A) To grasp theorem **2.6** let us start with finite vector spaces, which are close to usual vector spaces. E.g. on $GF(3)$ consider the linear transformation $x' = 2x$, $y' = 2x + z$ and $z' = 2x + y$ and the vectors $\mathbf{w}_0 = [0, 1, 1]$, $\mathbf{w}_1 = [1, 0, 1]$ and $\mathbf{w}_2 = [1, 1, 0]$, which are preserved up to the multipliers 1, 2, and 2 respectively. If V denotes our vector space, these three "innervectors" identify an innergenerator family $w: 3 \to \mathsf{F}1V$ of three singleton innerfamilies, viz. $wj0 = \mathbf{w}_j$ with $j = 0, 1, 2$, while the multipliers identify the corresponding innermorphisms. Then, by **2.6** blindly follow the familiar general integration procedure for complex vector spaces and get the general integral $v_i = c_0\mathbf{w}_0 1^i + c_1\mathbf{w}_1 2^i + c_2\mathbf{w}_2 2^i$, where by our vector space notation the algebraic (i.e. linear) form g is identified by the "arbitrary constants" c_j. From this get single integrals as usual, e.g. $v'_0 = [2, 0, 0]$ yields $v'_m = [2^{m+1}, 2^m + 2, 2^m + 2]$.

(B) To taste the difference between **2.6** and its extension **2.7**(B), consider the finite Boolean set algebras again. While the starting theorem concerns clocks as in **2.5**, its extension concerns any binary sequential circuit (where I is the set of inputs).

The Boolean case is interesting also because it allows us to compare our integration method and the structure theories of the 60's for sequential machines [19] and for universal algebras (direct and subdirect products [17] or other structures [31].) In general this is not possible, since their relevant objects (endomorphisms and algebras) are different. On the contrary, in the Boolean case, we can use the general integral to get "decompositions" of the clocks (or sequential circuits) concerned, since they are often thought of as singleton (or unary) algebras. In fact, it provides us an integral for each bit, i.e. for each $v_0 = gx$, g being the associating family. Clearly, such a family of integrals $\int: X \to \mathsf{F}NA$ gives us M again, after evaluation, $\mathbf{C}\!\int\!1 = M$. Hence, the parameters p_x corresponding to $\int x$ are the Boolean polynomials defining another circuit realization.

E.g., for the clock "1 0 3 2" of **2.1** choose the innergenerator family corresponding to the two innerelements $\{0, 2\}$ and $\{0, 3\}$. Then, our method provides us two Boolean polynomials $p_0 = \overline{x}$ and $p_1 = x \oplus y$ that realize the clock by two flip-flops (both innervalues were 0) instead of two delays.

If we should face such a problem by the structure theory of sequential machines, then we should work with 2^n states instead of our n bits. Therefore, some information concerning the cospectrum of our objects can allow us an exponential work reduction in the Boolean case.

Such a worth of cospectral information leads us to guess it is an expensive knowledge. This is confirmed by the next example concerning an even simpler class of algebras.

(C) Another case for tasting the difference between **2.6** and **2.7**(B) is the one of word catenation monoids. While the theorem concerns D0L systems (the simplest of the rewriting systems introduced in theoretical biology [36]), its extension concerns DT0L systems (another rewriting system [8] driven by a "control word", which becomes our integration variable).

The "sets of productions" of these systems are our families M, provided that we choose the most natural associating family g. In fact, from an endomorphism h of the catenation monoid on a set of words $A = X^*$ we get such an $M\colon X \to A$ by $Mx = h\{x\}$ for all $x \in X$, viz. g is the partition function of the identity on X. E.g. take the D0L system with productions $a \to a$ and $b \to baab$ that starts with the "axiom" $v_0 = aaab$, where the words on $X = \{a, b\}$ have been denoted by the usual Formal Language notation. Then, the words $w00 = a$ and $w10 = aab$ define a $w\colon 2 \to \mathsf{F}1A$, which clearly is an innergenerator family for the submonoid they generate. Since this submonoid contains v_0, we can use our procedure on it and get the "word sequence" $v_n = a(aab)^{2^n}$.

In order to find a suitable general integral for a given set of productions we have to search for innerspaces. Yet, the innerlattice of an endomorphism for word catenation is known as a tricky closure system. Just to decide whether the closure of a word is catenationally generated by some finite innerfamily or only by infinite ones is a non trivial problem. In fact, it is an instance of the (open) LCF decidability problem, as shown in **1.2.3** of [36].

3 Forms of higher degree.

3.0 Definitions. The *power of* an algebra α on A *to* a set J is the algebra $\mathcal{F}J\alpha$, on the set of the J*–families* $\mathsf{F}JA$, defined by

$$\mathcal{F}J\alpha\sigma f = \alpha\sigma \cdot \mathbf{C}f \quad , \text{ for all } \sigma \in \Sigma \quad \text{and } f\colon S\sigma \to \mathsf{F}JA \; . \tag{19}$$

Given α and a set of unknowns X, their *connection algebra* is the algebra ϑ on $\mathsf{H}\tau\alpha$ defined through (10) by

$$\vartheta\sigma l = \mathbf{C}\overline{\alpha}(\alpha\sigma \cdot \mathbf{C}l \cdot \mathbf{i}_X) \quad , \text{ for all } \sigma \in \Sigma \quad \text{and} \quad l\colon S\sigma \to \mathsf{H}\tau\alpha \; . \tag{20}$$

This is to say that ϑ acts the homomorphisms of $\mathsf{H}\tau\alpha$ only through their representations by "allocations to the unknowns" and its operations act these families of allocations componentwise in α.

3.1 Lemata. (A) Given α and J as above and any algebra β, consider the function C such that $Ch = \mathbf{C}h$ for any homomorphism family $h\colon J \to \mathsf{H}\beta\alpha$. *Then,*

$$C : \mathsf{F}J(\mathsf{H}\beta\alpha) \rightarrowtail\!\!\!\twoheadrightarrow \mathsf{H}\beta(\mathcal{F}J\alpha) \quad . \tag{21}$$

(B) *Given X as above, $\mathbf{C}\overline{\alpha}$ is an isomorphism between $\mathcal{F}X\alpha$ and ϑ.*

Proofs. (A) Clearly, C is one to one. Hence, we only have to check the codomain, i.e. to prove for all $b : S\sigma \to B$ that $\mathbf{C}h(\beta\sigma b) = \mathcal{F}J\alpha\sigma(\mathbf{C}h \cdot b)$ iff $hj(\beta\sigma b) = \alpha\sigma(hj \cdot b)$ for all $j \in J$. (In fact, by (2) any $k \in \mathsf{H}\beta(\mathcal{F}J\alpha)$ is a $\mathbf{C}h$, for $h = \mathbf{C}k$.) This equivalence easily follows from (2) and (3) after applying j to both sides of the former equality.

(B) After (10), we only need to prove the homomorphic condition $\mathbf{C}\overline{\alpha}(\mathcal{F}X\alpha\sigma f) = \vartheta\sigma(\mathbf{C}\overline{\alpha} \cdot f)$, where $f\colon S\sigma \to \mathsf{F}XA$. By (19) and (20) this is true whenever $\mathbf{C}f = \mathbf{C}(\mathbf{C}\overline{\alpha} \cdot f) \cdot \mathbf{i}_X$. This latter equality easily follows from (6) through $\mathbf{C}$–reduction by applying any $x \in X$ and $s \in S\sigma$ to both sides. *Q.D.E.*

3.2 Corollary. *X–anarchic extensions preserve powers,*

$$\overline{\mathcal{F}J\alpha} = \mathcal{F}J\overline{\alpha} \quad . \tag{22}$$

Proof. It is enough to prove that $\mathbf{C}\overline{\mathcal{F}J\alpha} = \mathbf{C}(\mathcal{F}J\overline{\alpha})$. Note that both sides have values in $\mathsf{H}\tau(\mathcal{F}J\alpha)$: the left one by (10) and the right one by (21), where $\beta = \tau$, since, for all $a : X \to \mathsf{F}JA$, $\mathbf{C}(\mathcal{F}J\overline{\alpha})a$ equals a $\mathbf{C}h$ with $h = \mathbf{C}\overline{\alpha} \cdot \mathbf{C}a : J \to \mathsf{H}\tau\alpha$ by (10). (Apply $t \in T$ and use $\mathbf{C}$–reduction, (19) and (3).) Hence, by the inverse of (10) we can prove our equality after composing its sides with $\mathbf{ri}_X$. This is to say we only need to prove (22) with arguments in X. That is quite easy by (6) and $\mathbf{C}$–reduction. *Q.D.E.*

3.3 Definitions. Let X be a set of natural numbers. (Clearly, this is a set of unknowns for all S.) Then, we say that the terms in T are *symbolic* $\mathbf{i}$*–ary forms of higher degree on* (or *with powers in*) X. ("$\mathbf{i}$–ary" stands for "of one variable". More variables are considered in **3.9**(A).) Given an algebra α on A, consider its power to A, $\varphi = \mathcal{F}A\alpha$. For each $h \in \mathsf{H}\alpha\alpha$, let $h'\colon X \to \mathsf{F}AA$ be the family of composition powers $h'x = h^x$, for all $x \in X$, and set $h'' = \mathbf{C}h'\colon A \to \mathsf{F}XA$. Then, define $\widetilde{\varphi}\colon T \to \mathsf{F}(\mathsf{H}\alpha\alpha)(\mathsf{F}AA)$ by $\widetilde{\varphi}th = \overline{\varphi}th'$ for all $t \in T$ and $h \in \mathsf{H}\alpha\alpha$ and call any $\widetilde{\varphi}t$ an *(*$\mathbf{i}$*–ary) form of higher degree of* α (see **3.9**(A)). Two symbolic $\mathbf{i}$–ary forms of higher degree t and d are called *equivalent*, $t \equiv d$, when $\widetilde{\varphi}t = \widetilde{\varphi}d$.

Since by (10) each $k = \mathbf{C}\widetilde{\varphi}h$ belongs to $\mathsf{H}\tau\varphi$, we call k the *C(ayley)–H(amilton) homomorphism for* α *and* h. If $kt = kd$ with $t, d \in T$, then we say that h *satisfies the scalar equation* $\langle t, d\rangle$ (or $t = d$). Thus, a scalar equation satisfied by h is not algebraic for h, viz. we use h', not h, in (10) to get our k. We denote by E' the set of all scalar equations satisfied by h.

Given t, $d \in T$, consider an eigengenerator G for h. If, for each eigenspace in G, its eigenmorphism m satisfies $t = d$, then we say that $t = d$ is a *(weakly) characteristic equation of* h *at* G. ("Weak" refers to the fact that even other endomorphisms are allowed to satisfy it. Weak equations will allow the universal Cayley–Hamilton theorem to have a form stronger than the Linear Algebra one.) $E''G$ denotes the set of all such equations.

3.4 Lemata. *Composition powers preserve eigenpartitions and innerfamilies, viz., given* h, m *and* n *as in* **1.6** *and any natural number* x, m^x *and* n^x *are an eigenmorphism and an innermorphism respectively for* h^x.

Proofs. (Eigen) The statement

$$p \cdot h^x = m^x \cdot p \tag{23}$$

is trivial, when $x = 0$. The remaining induction step follows from (11) after the compositions of h with each side of (23).

(Inner) Our statement now is

$$h^x \cdot v = n^x \cdot v \tag{24}$$

and has a proof similar to the one for (23), but for a reversal of the h composition and for the use of (13) instead of (11). *Q.D.E.*

3.5 Theorems. (A) *An endomorphism satisfies a scalar equation iff the equation is characteristic, viz.* $E' = E''G$ *for any eigengenerator* G. (B) *A characteristic equation is satisfied by all innermorphisms.*

Proofs. (A) From (23) it follows that

$$\mathbf{B}_X p \cdot h'' = m'' \cdot p \quad , \tag{25}$$

as one could easily check by applying any $a \in A$ and any $x \in X$ to both sides. Now, by (22) h satisfies $t = d$ iff

$$\overline{\alpha}t \cdot h'' = \overline{\alpha}d \cdot h'' \quad . \tag{26}$$

Given any eigenspace $\gamma \in G$, compose both sides of the latter equality with the corresponding p. Then, by (9) and (25) we get $\overline{\gamma}t \cdot m'' \cdot p = \overline{\gamma}d \cdot m'' \cdot p$. Hence, since p is onto Q, if h satisfies $t = d$, then by **3.2** any such m does. Conversely, if all m's do, then h does, because we can reverse these passages and get the identity partition on A from the infimum of our p's.

(B) From (24) and (13) it easily follows that $n'' \subseteq h''$, for any innermorphism $n \in H\delta\delta$. Hence, if we compose $\mathbf{i}_{\mathbf{c}v}$ and each side of (26), then we get $\overline{\delta}t \cdot n'' = \overline{\delta}d \cdot n''$, i.e. by **3.2** n satisfies $t = d$. *Q.D.E.*

3.6 Examples. (A) Once we restrict our universal Cayley–Hamilton theorem to specific classes of algebras, can it tell us non trivial properties of specific kinds of objects ? Let us check it in a worst case, i.e. by algebras definitely more trivial than vector spaces.

Consider the algebras of complete set union. (In the finite, to get all unions we just need an algebra consisting of binary union and the empty constant. In the infinite case as well as in general, a single "P–operation" [32], albeit not an $\mathsf{F}r$–one, suffices, see **4.6** ibidem. "P–algebras" do not differ from the present ones for our purposes, see **1.8** ibidem and **6.9**(B). Anyway, we could also resort to an infinity of $\mathsf{F}r$–operations as in [24] with no big loss of triviality.)

As shown in **I.3.5** of [24], the real objects of our algebras, viz. the M's of **1.6**, are (directed) graphs, i.e. relations. From our definitions, the equations satisfied by a graph are identified by couples of sets of natural numbers n and represent the equivalent sets of n–steps spans in the graph. In fact, h^n is associated with the n–th relational composition power of relation M.

Clearly, these couples are invariant with respect to graph isomorphism and it is fairly trivial that they are satisfied by any innermorphism, e.g. within the weakly connected components generated by some vertices of the graph. It is less trivial that they are satisfied by an eigenmorphism, which can be a contracted graph, or that the couples for any eigengenerator are the same ones of the graph.

(B) We cannot strengthen **3.5**(B) into a "dual" of **3.5**(A). Namely, *in general it is false that an endomorphism has to satisfy an equation that is satisfied by all innermorphisms of an innergenerator*. This is proved by the following counterexample, where we use the notation and notions of [36].

Consider the D0L system with $\omega_0 = ab$ and with productions $a \to aa$ and $b \to bb$. Clearly, $\omega_1 \neq \omega_0\omega_0$. Hence, the endomorphism associated with our productions does not satisfy the equation $h(w) = ww$. Yet, this equation is satisfied whenever w is on a singleton alphabet, i.e. it is satisfied by both innermorphisms of the innergenerator with the two singleton–alphabet innerspaces.

3.7 Definitions. Given S and X as above, a function $u: A \to \mathsf{F}TA$, such that $u = \mathbf{C}\overline{\alpha}\cdot h''$ for some algebra α on A of species S and some $h \in \mathsf{H}\alpha\alpha$, is called *a retro-operation on* A (or *the retro-operation of* α *and* h) and, for each α, $j\alpha: \mathsf{H}\alpha\alpha \to \mathsf{F}A(\mathsf{F}TA)$ denotes the function such that $j\alpha h = u$ for all h's. U denotes the class of all retro-operations and U' denotes the corresponding class of C–H homomorphisms $u': T \to \mathsf{F}AA$ as in **3.3** for each our α and h. For $u',v' \in U'$, $\mathsf{H}'u'v'$ denotes the set of *homomorphisms from* u' *to* v' defined as the homomorphisms between the corresponding algebras of a singleton valued species on T (e.g. u' corresponds to a $u'': T \to \mathsf{F}(\mathsf{F}\{\emptyset\}A)A$, with $u''ta = u't(a\emptyset)$ everywhere). We will also say that u is *on* A.

Given $f\colon A \to B$, we denote by $\mathbf{e}f$ its $\mathbf{S}F(FT)$*–extension*, which we define as the function $\mathbf{e}f\colon FA(FTA) \to P(B \times FTB)$ such that, for all $g: A \to FTA$, $\mathbf{e}fg = \{\langle fa, f \cdot ga\rangle \mid a \in A\}$. If

$$\mathbf{e}fu \subseteq v \quad , \tag{27}$$

for some u as above and $v \in U$ on B, then we call f a *rheomorphism from* u *into* v (of *aylotype* $\mathbf{S}F(FT)$) and Ruv denotes the set of all such f's. (See the meaning of "$\mathbf{S}F(FT)$" in **6.8**.) If f is onto B, i.e. $\mathbf{e}fu = v$, then we say that f is *onto* v.

When X contains all natural numbers, we define the *shift* as the endomorphism $s = \mathbf{C}\overline{\tau}\mathbf{s} \in H\tau\tau$, $\mathbf{s}\colon X \to T$ being the successor function of natural numbers. Then, we define the *anarchic* retro-operation as $r = j\tau s$, while we call *anarchic* its corresponding C–H homomorphism $\mathbf{C}\widetilde{\mathcal{F}T}\tau s$. For each $u = j\alpha h$, we denote by ρ the (connection like) algebra on Rru such that $\rho\sigma l = u(\alpha\sigma(\mathbf{C}l0))$ for all $l: S\sigma \to Rru$ and $\sigma \in \Sigma$, see (20).

Since $r\colon T \to FTT$, we define the *anarchic convolution* as the binary operation $\bullet: T \times T \to T$, with infix notation, such that $t \bullet d = rdt$ for all $t, d \in T$. Viz. $t \bullet d$ is a formal multiplication of t and d, since $rdt = \overline{\tau}t(s''d)$ is the term we get from t when d, shifted x times, replaces each "leaf" $x \in X$. (In vector spaces, "convolution" is used in the infinite dimensional case, otherwise "polynomial multiplication" is used. Here, we cannot use the latter (correct) word, since we accepted the conventional (mis)use of "polynomial" in Universal Algebra.) The *convolution of two forms of higher degree* f *and* g *of* α is the family $\Phi\mathbf{B}fg\colon H\alpha\alpha \to FAA$, defined by $\Phi\mathbf{B}fgh = fh \cdot gh$ for all $h \in H\alpha\alpha$.

3.8 Theorems. Keep the previous notation. (A) $f \in Ruv$ *iff*

$$\mathbf{B}_T f \cdot u = v \cdot f \quad . \tag{28}$$

(B) *Let* C *be the function on* U *such that* $Cu = \mathbf{C}u$ *everywhere. Then* $C\colon U \mapsto\!\!\twoheadrightarrow U'$, *where* Cr *is anarchic (i.e. when* $\alpha = \tau$, $\mathbf{C}r = \mathbf{C}\widetilde{\varphi}s$*) and* (C) $Ruv = H'(Cu)(Cv)$. (D) *When* $h \in H\alpha\alpha$ *and* $k \in H\beta\beta$, $f \in H\alpha\beta$ *and* $f \cdot h = k \cdot f$ *imply* $f \in R(j\alpha h)(j\beta k)$, *while the converse does not hold.*

(E) *If* $f \in Ruv$ *and* $g \in Rvw$, *then* $g \cdot f \in Ruw$. (F) *If* u *is on* A, *then* $\mathbf{i}_A \in Ruu$. (Hence, composition on Ruu and $\mathbf{i}_A$ form a monoid.)

(G) *If* $f \in H'uv$ *is onto* v, *then*

$$Eu \subseteq Ev \quad , \tag{29}$$

where Ew, *with* $w = u, v$, *denotes the set of (characteristic) equations induced by* w. (H) *When* $f \in H'vu$ *is one to one, (29) still holds.*

(I) *For each* α, $j\alpha\colon H\alpha\alpha \to H\alpha\vartheta$. *Moreover,* (J) *when* X *contains all natural numbers,* $j\alpha$ *is one to one and* (K), *for all* $u = j\alpha h$,

$$(30) \qquad u\colon A \Vdash\!\!\twoheadrightarrow Rru$$

is an isomorphism from α *to* ρ, *with* $u^{-1} = \mathbf{Ci}_{Rru}0$. (Hence, each ρ is a subalgebra of ϑ.)

(L) r *is an isomorphism from anarchic convolution onto the composition on* Rrr *as in (E),* $r(t \bullet d) = rd \cdot rt$, *for all* $t, d \in T$. (Hence, anarchic convolution forms a monoid.) More in general, (M) *given* α *on* A, *for each* C*–*H *homomorphism* $k = \mathbf{C}\widetilde{\varphi}h$, *composition on* $k^{\uparrow}T \subseteq FAA$ *forms a monoid, which is the homomorphic image of the converse anarchic convolution monoid under* k, *i.e. for all* $t, d \in T$,

$$(31) \qquad k(t \bullet d) = kt \cdot kd \quad .$$

(Hence, the convolution between forms of higher degree defines a monoid and equivalence between symbolic **i**–ary forms of higher degree is a congruence of anarchic convolution.)

Proofs. (A) If we apply any $a \in A$ to both sides of (28), then we get the statement $\langle fa, f \cdot ua \rangle \in v$, which proves (27), and conversely.

(B) Clearly, C is one to one and, for each α, every $\mathbf{C}\widetilde{\varphi}h \in U'$ has an h that determines it. Hence, it is enough to prove that, for each $u = j\alpha h$, $Cut = \mathbf{C}\,\overline{\mathcal{F}A\alpha}\,h't$ for all $t \in T$. Both sides equal $\overline{\alpha}t \cdot h''$, the left one by (3) and (2) and the right one by (22) and $\mathbf{C}$–reduction. This also shows the preservation of anarchy.

(C) It is enough to show that

$$(32) \qquad f \cdot Cut = Cvt \cdot f \ , \quad \text{for all } t \in T,$$

is equivalent to (28) or, by $\mathbf{C}$–reduction, to $\mathbf{C}(\mathbf{B}_T f \cdot u)t = \mathbf{C}(v \cdot f)t$ for all $t \in T$. The right sides are equal by (3), while the left ones are found equal by (3) after applying any $a \in A$.

(D) As for (23), we easily get $f \cdot h^x = k^x \cdot f$ for all natural numbers x. Hence, as in (25) $\mathbf{B}_X f \cdot h'' = k'' \cdot f$ and from (9) we get (32) by $Cut = \overline{\alpha}t \cdot h''$. Therefore, (28) is satisfied.

In order to prove that $R(j\alpha h)(j\beta k)$ can properly contain the set of f's satisfying our premises, take $\alpha = \beta$ being the real vector plane and $h = k$ being the identity. Then, a nonlinear f, satisfying (32) viz. $f \cdot l = l \cdot f$ for all $l = \overline{\alpha}t \cdot h''$ can be easily found, since l always is a dilatation. E.g. such an f is given by the real system $x' = x$ and either $y' = 0$, when $x = 0$, or $y' = y^2/x$ otherwise.

(E) and (F) Transitivity and reflexivity of set containment.

(G) Let $f: A \twoheadrightarrow B$. Choose any function $g: B \to A$ such that $f \cdot g = \mathbf{i}_B$ and define $l: \mathsf{F}AA \to \mathsf{F}BB$ by $la = f \cdot a \cdot g$ for all $a: A \to A$. Since $f \cdot ut = vt \cdot f$ for all $t \in T$, we easily see that $l \cdot u = v$, by applying t to both sides, which implies (29).

(H) Dual of (G) (viz. $f: B \rightarrowtail A$, $g: A \to B$ with $g \cdot f = \mathbf{i}_B$ and l, such that $la = g \cdot a \cdot f$, yield $l \cdot u = v$, since $f \cdot vt = ut \cdot f$).

(I) Clearly, $h^x \in \mathsf{H}\alpha\alpha$ for all $x \in X$ and $h \in \mathsf{H}\alpha\alpha$. Hence, by (21) $h'' \in \mathsf{H}\alpha(\mathcal{F}X\alpha)$. Therefore, our statement follows from **3.1**(B) and **1.1**(C).

(J) Since, $1 \in X$, by (3), (2) and (6) $h = \mathbf{C}(j\alpha h)1$, which implies our injectivity.

(K) We first show that, for all $l \in \mathsf{H}\tau\alpha$,

$$(33) \qquad l \cdot s = h \cdot l \quad \text{iff} \quad l = ua \text{ , for some } a \in A.$$

To get the "if", by the inverse of (10) it is enough to show that, when $l = \mathbf{C}\overline{\alpha}(h''a)$, $l(sx) = h(lx)$ for all $x \in X$. (In fact, both sides of the premise in (33) are in $\mathsf{H}\tau\alpha$ by (10) and **1.1**(C).) By **C**–reduction and (6) this is to say that $h^{x+1}a = h(h^x a)$, i.e. a triviality. To get the "only if", set $a = l0$ and $l' = l \cdot \mathbf{i}_X$. Then, by (10) $l = \mathbf{C}\overline{\alpha}l'$, where $l' = h''a$ by induction on the premise. Hence, $l = ua$.

Now, by (D) $ua \in \mathsf{R}ru$ for all $a \in A$. In fact, by (10) $ua \in \mathsf{H}\tau\alpha$ and we got (33). Also, u has to be one to one, since $ua0 = a$ for all $a \in A$. Moreover, $f \in \mathsf{R}ru$ by (32) implies both $f \cdot s = h \cdot f$, since $s = \mathbf{C}r1$ and $h = \mathbf{C}u1$, and $f \in \mathsf{H}\tau\alpha$, since $\overline{\tau}t(s''0) = t$ (after applying $0 \in T$ to both sides of $f \cdot \overline{\tau}t \cdot s'' = \overline{\alpha}t \cdot h'' \cdot f$, abstract t by **C**–reduction and use (10)). Hence, u in (30) is onto and $ua = f$ for $a = f0$, which implies the inverse we claimed.

It remains to show that $u(\alpha\sigma a) = \rho\sigma(u \cdot a)$ for all $a : S\sigma \to A$ and $\sigma \in \Sigma$. The right side becomes $u(\alpha\sigma(\mathbf{C}u0 \cdot a))$ by (3) and equals the left one, since $\mathbf{C}u0 = \mathbf{i}_A$.

(L) Take $u = r$ in (K). Then, we only need to prove that $r(rtd) = rt \cdot rd$ for all $t, d \in T$. Since both sides are in $\mathsf{R}rr$, by the inverse of (30) we only have to check their equality at $0 \in X$. This easily follows from (6).

(M) It is enough to prove (31). Apply any $a \in A$ to both sides. Let u denote $\mathbf{C}k = \mathbf{C}\overline{\alpha} \cdot h''$. Then the left side by **C**–reduction yields $ua(rdt)$, while the right one yields $u(uad)t$. By (B), (A) and **B**–reduction this is to require that $ua \in \mathsf{R}ru$. Hence, (30) implies (31). *Q.D.E.*

3.9 Comments. (A) So far, we have universalized the treatment of vector space polynomials with one variable. (Only such polynomials are needed in the classical Cayley–Hamilton theorem.) Now, we consider the case with more variables. It enables us to see some algebraically uncommon features of our theory more clearly. In (B), it will link our theory and the one of quasiendomorphisms. Furthermore, in (C) we will show that the "symbolic forms of higher degree" (the extension of symbolic $\mathbf{i}$–ary forms of higher degree to the more variables case) can formalize the syntax of computer programs better than the derivation trees of Formal Language Theory [0] can.

In **3.3** X was the set (or a subset) of natural numbers, used as exponents for the composition powers of a single endomorphism (one variable case). In the case of more variables, we introduce these variables as the letters of a primary alphabet I as in **2.7**(A). Then, we take X as the set or a subset of words on I (in order to keep track of the various endomorphisms in a composition) and the remainder will closely follow the one variable case. While φ in **3.3** is kept, h becomes a family, $h: I \to \mathsf{H}\alpha\alpha$. Hence, any $h'x$ is the $\odot$ composition as in **2.7**(B) of the h_i's corresponding to x, h' and h'' keep their old types and $\widetilde{\varphi}: T \to \mathsf{F}(\mathsf{FI}(\mathsf{H}\alpha\alpha))(\mathsf{F}AA)$. Therefore, any *form of higher degree of* α now is of the type $\widetilde{\varphi}t: \mathsf{FI}(\mathsf{H}\alpha\alpha) \to \mathsf{F}AA$, where $\mathsf{H}\alpha\alpha$ is prefixed by FI. Thus, we might also call it an FI*–ary form of higher degree* as opposed to the **i**–ary forms of **3.3**.

We immediately find that the definitions of C–H homomorphism, retro-operation and rheomorphism do not change (but for the new type of $j\alpha: \mathsf{FI}(\mathsf{H}\alpha\alpha) \to \mathsf{F}A(\mathsf{F}TA)$). On the contrary, the shift becomes a family $s: I \to \mathsf{H}\tau\tau$, where $s_i = \mathbf{C}\overline{\tau}(\varsigma i)$ for all $i \in I$, $\varsigma: I \to \mathsf{F}I^* I^*$ being the (unary) anarchic algebra of the primary alphabet. Consequently all the anarchic stuff (retro-operation, C–H homomorphism and convolution) as well as algebras ρ keep their old types. (In the definition of ρ, 0 is the empty word and, in the one of r, the iterated shifts become word catenations.) Finally, the convolution of two forms of higher degree changes its type into $\Phi\mathbf{B}fg: \mathsf{FI}(\mathsf{H}\alpha\alpha) \to \mathsf{F}AA$.

Some of the statements of **3.8** are relevant to the following, once they are extended to FI–ary forms of higher degree. Relevant statements are **3.8**(A), (B), (C), (E) and (F), which concern the categories on U and U' and hold without any change for any subset X of words we choose. (Their proofs stay unchanged.) Other statements require minor changes, e.g. in **3.8**(D) we assume $f \cdot h_i = k_i \cdot f$ for all $i \in I$ with present h and another family $k: I \to \mathsf{H}\beta\beta$ and we keep an almost unchanged proof.

There also are relevant statements concerning the only case of the whole set X of the words on I. Consider the basic assertion (30) stating the freedom of r. We can check that (30) holds in the extended FI–ary case. In fact, if we change the premise of (33) into "$l \cdot s_i = h_i \cdot l$ for all $i \in I$ ", then we can again derive its "if" part from $h'(\varsigma ix)a = h_i(h'xa)$, while the "only if" stays as it was. From this new (33) we easily get (30) (unchanged) by few changes in the old proof (use $s_i = \mathbf{C}ri$ and $h_i = \mathbf{C}ui$). After extending (30), we immediately get the extensions of **3.8**(K), (L) and (M). In fact, the corresponding old proofs need no change.

The theory, we have found, looks very close to an "algebraic theory" [24]. Yet, there are not "insertion–of–the–variables maps" suitable to a clone composition as our convolution and to our primary variables (or to X). (r, $\mathbf{C}r$ and $\bullet$, as well as T, are determined by I through X. Hence, we should take the monoid on I^* as the typical object of our base category. Now, check the two axioms for algebraic theories in clone form as in **1.3.3** of [24]. The former requires that any insertion–of–the–variables map be a constant with the identity form as its value, whereas, in general, such a constant map cannot satisfy the latter, even in vector spaces.)

Moreover, the convolution of two forms "on" two I's is on their union. Hence, as the conventional monoidal categories of Algebra deserve the name of algebraic theories, so the theory of the forms of higher degree should be considered nonalgebraic.

Even when we disregard this, monoidal categories do not work here, because of the several links between endomorphisms, algebraic forms and forms of higher degree. The first link is the Cayley–Hamilton theorem, shown to be universal in **3.5**(A). Another is the category containment, implicit in **3.8**(D). Furthermore, there are other useful links. E.g. in vector spaces we can exploit the right distributivity: $(\sum_i c_i p_i(x))q(x) = \sum_i c_i p_i(x)q(x)$, where the $p_i(x)$'s and $q(x)$ are polynomials of one variable. Now, we can easily see that this property holds in any algebra and also that its proof is a special case of the associativity property of the (anarchic) clone composition.

(If we have $t \in T_Y$, $t' : Y \to T_X$, $t'' : X \to T_X$ and think of t as an element of a family of such terms, then the above mentioned associativity ensures us that $\overline{\tau}^X_X(\overline{\tau}^Y_X tt')t'' = \overline{\tau}^Y_X t(\mathbf{C}\overline{\tau}^X_X t'' \cdot t')$. Now, take $t'' = s''d$ where s is the shift, $d \in T_X$ and $X = N$. Then, we can reread our associativity derived equality as $\overline{\tau}^Y_X tt' \bullet d = \overline{\tau}^Y_X t(t' \otimes d)$ where $t' \otimes d : Y \to T_X$ is such that $(t' \otimes d)y = t'_y \bullet d$. Note that $\overline{\tau}^Y_X t$ (as in vector spaces $\sum_i c_i$) gives us an algebraic (linear) form, while the t'_y's and d play the same rôle of the former p_i's and q. Once we have got this anarchic distributivity, **3.8**(M) extends it to any algebra.)

Another (minor) universal link is that the forms of higher degree are an "extension" of algebraic forms, as they are in vector spaces. Formally, let $\widetilde{\varphi}t : \mathsf{F}I(\mathsf{H}\alpha\alpha) \to \mathsf{F}AA$ be of degree one, i.e. let t be on $X = I^1$. Then, for all such t's, $\widetilde{\varphi}t$ behaves as the algebraic form $\overline{\varphi}t$, i.e. $\widetilde{\varphi}t = \overline{\varphi}t \cdot \ell$, where $\ell : \mathsf{F}I(\mathsf{H}\alpha\alpha) \mapsto\!\!\twoheadrightarrow \mathsf{F}I^1(\mathsf{H}\alpha\alpha)$ is the relabelling such that $\ell h\langle\emptyset, i\rangle = hi$ everywhere.

As in Algebra one does not consider a ring as a group nor as a semigroup, so here we can hardly consider our theory as an algebraic theory formalized by a (monoidal) category.

(B) When h is a family of endomorphisms of α, we can relate our forms of higher degree with the "quasiendomorphisms" of [30]. In fact, when h indexes a set of generators of the endomorphism monoid of α, we get any quasiendomorphism of α from some form of higher degree $\widetilde{\varphi}t$ by applying h. Conversely, given any $\widetilde{\varphi}t$, its value $\widetilde{\varphi}th$ is a quasiendomorphism, i.e. the function $\mathbf{C}(j\alpha h) = \mathbf{C}\widetilde{\varphi}h : T \to \mathsf{F}AA$ is onto the set $Q \subseteq \mathsf{F}AA$ of the quasiendomorphisms.

The wealth of properties in **3.5**, **3.8** and **3.9**(A) rises from our wider freedom of choosing h. This freedom allowed us to "structure" (the abstraction of) quasiendomorphisms into the forms of higher degree through the indexing by h. Yet, when we disregard the type distinction from a $q \in Q$ and any $\widetilde{\varphi} th \in (\mathbf{C}\widetilde{\varphi} h)^{\uparrow} T \subseteq Q$, i.e. under the Combinatory Logic point of view, the link between our forms and quasiendomorphisms is tight, since it is close to an "application–abstraction" link.

(C) Formal Language Theory [0] formalizes many high level programming languages by "context free grammars". A "derivation tree" of such a grammar specifies the syntactic structure of a source program that a translator has to follow during program compilation or interpretation [1]. Such a formalization succeded both as a practical theory for programmers writing translator programs and as a well defined method for writing programs generating (parts of) translator programs (once the former programs are fed with syntactic specifications of the programming language considered [18], [1]). These "translator writing systems" solved many reliability problems felt in the software industry.

In spite of these successes, the match between a (context free) formal language and its programming language is not perfect. A first mismatch appears when we consider a real translator. There, the processing of a source program begins with two logically distinct steps: lexical analysis and syntactic analysis. Both of them rely upon Formal Language Theory. Yet, their distinction does not, since Formal Language Theory allows us a unified analysis step.

A closer view at the (real life) dichotomic analysis reveals the typical structure we would expect when processing our symbolic forms of higher degree. During lexical analysis we scan an input sequence on a "primary" alphabet I in order to get a set X of words ("tokens") on I. Set X provides the translator with (more or less structured) "tables" serving both the syntactic analysis and further (semantic) steps. During syntactic analysis we build syntactic structures starting from X, while disregarding the input alphabet.

On the contrary, the usual derivation trees of Formal Language Theory, as well as the (logically better) trees of [41], are a case of the terms of section **1**. In fact, they correspond to symbolic polynomials of conventional universal algebras.

Another mismatch between programming languages and formal (context free) languages is the lack of context freedom in the (many) programming languages where an identifier has to be declared. (Such a restriction is a well known case of lacking context freedom.) However, if we extend the notion of a context free language from symbolic polynomials (terms) to symbolic forms of higher degree, then we have to consider the syntactic analysis only, viz. we consider τ_X, not a τ_I. Therefore, our mismatch vanishes, since the identifiers belong to X. (See also **6.1**.)

4 Normal characteristic equations.

4.0 Definitions. Let X be the set of all natural numbers, α be as in **3.3** and $h \in H\alpha\alpha$ be associated with an $M\colon Y \to A$. Then, we call a characteristic equation of h *normal*, when it has the form $\langle n, t\rangle$ with $t \in T_n$ and $n \in X$ is minimal, viz. if $\langle m, d\rangle$ is a characteristic equations with $d \in T_m$, then $n \leq m$. (Recall that $0 = \emptyset$ and $n+1 = \{0, \ldots, n\}$ by **0.6**.) Also, we respectively call n and t the *degree* of our equation and a *characteristic term* of M or of h.

We say that $t \in T_X$ has *degree* n, when $t \in T_{n+1}$ and n is minimal. As usual, we say that our species S is *finitary*, when $S\sigma$ is finite for all $\sigma \in \Sigma$. Clearly, all t's have degrees iff S is finitary.

Let k be the C–H homomorphism for α and h and let n be a natural number. If for each $t \in T_X$ there is a $d \in T_n$, such that $kt = kd$, and n is minimal, i.e. either $n = 0$ or $k^{\uparrow}T_{n-1} \subset k^{\uparrow}T_X$, then we say that n is the *dimension of* h (or *of* M *with respect to* the corresponding associating family and algebra) and that h is *finite dimensional.* (E.g. the dimension of a dilatation in a vector space of any dimension is one.)

4.1 Theorem. *When the species is finitary, an endomorphism has a normal characteristic equation iff it is finite dimensional. Also, the degree of such an equation equals the dimension of the endomorphism. In general, a finite dimensional endomorphism has a normal characteristic equation.*

Proof. (Only if) Let $\langle n, t\rangle$ be normal characteristic and m be the degree of any $t' \in T_X$. Then, it is enough to show that, when $m > n$, there is a $d \in T_m$ such that

$$kt' = kd\ , \tag{34}$$

k being our C–H homomorphism. Our normal equation will perform such a degree reduction.

Consider the family $f\colon m+1 \to T_m$ such that $fx = x$ for $x < n$ and $fx = t \bullet (x-n)$ elsewhere. Then, all the degrees of the fx's are less than m and, since the endomorphism satisfies $\langle n, t\rangle$, (31) yields

$$k \cdot f = k \cdot \mathbf{i}_{m+1} \quad . \tag{35}$$

Now, set $d = \overline{\tau}_m^{m+1}\, t' f \in T_m$ and check (34). The right side becomes the left one, $\overline{\alpha}t' \cdot h'' = kt'$, by (9), (35) and (6), since k is a homomorphism from τ_X and by **1.1**(C) from τ_m too.

(If) Let n' be the dimension of the endomorphism. Then, $n' \in T_{n'+1}$ has a $t \in T_{n'}$ such that $kn' = kt$, i.e. $\langle n', t\rangle$ is characteristic. Since the set of such equations is not empty, there is a minimal degree equation in it, viz. a normal characteristic equation.

(Also) In order to see that our n and n' are equal, consider the reduction in the "only if" part, which implies $n \geq n'$, and the "if" part, which implies $n \leq n'$.

(In general) In the (If) proof we did not require a finitary S. *Q.D.E.*

4.2 Definitions. We say that a family and its associated endomorphism have a *null determinant*, when they have a characteristic term that is the shift of another term. As usual, we say that an algebra with no more than one element is *trivial*.

4.3 Theorems. *For all non trivial algebras* (A) *a one to one endomorphism cannot have a null determinant and* (B) *a null determinant endomorphism cannot be onto its domain.*

Proofs. (A) We prove it by contradiction and with the notation of **4.0**. Let $n = 0$. Then, since $kn = \mathbf{i}_A$ while kt with $t \in T_\emptyset$ has to be constant, we contradict the non triviality assumption.

Now let $n > 0$. Consider any $d \in T_X$ such that $sd = t$. Since $sd = 1 \bullet d$, $kt = h \cdot kd$ by (31). Then, since h satisfies $\langle n, t\rangle$, $h \cdot h^{n-1} = h \cdot kd$. Therefore, by the injectivity of h, $\langle n-1, d\rangle$ is a characteristic equation, contrary to the minimality of n.

(B) In the proof of (A) we reverse the factorization of h. I.e., when $n > 0$, from $kn = k(sd)$ we easily get $k(n-1) \cdot h = kd \cdot h$, since $sd = d \bullet 1$. Then, the surjectivity of h gives us a normal characteristic equation of degree $n - 1$. *Q.D.E.*

4.4 Comment. As shown in [26], "finite dimensional" universal algebras (unlike vector spaces) may have not dimensions. In fact there are algebras where an infinity of bases with different finite cardinalities coexist. However, such a finding only concerns "empty" universal algebras in the following sense.

As assumed in **0.2**, the only objects of our applications are families M corresponding to endomorphisms h, not the algebras themselves. Here, such true objects have been shown to have their (single) geometric dimensions exactly as the true objects of vector spaces do. To consider an algebra without such objects is not very appealing for our application purposes. Hence, algebra adimensionality might hardly appear as a calamity. See also **6.12**(A).

5 Bases.

5.0 Lemma. *Let α and β be algebras on A and B respectively and let X be a set of unknowns, as in* **1.2**. *Assume that β is connected with $b: X \to B$ and that*

$$\overline{\alpha} = \chi \cdot \mathbf{C}\overline{\beta}b \ , \quad \text{for some} \quad \chi : B \to \mathsf{F}(\mathsf{F}XA)A \quad . \tag{36}$$

Then,

$$\mathbf{C}\chi : \ \mathsf{F}XA \twoheadrightarrow \mathsf{H}\beta\alpha \quad . \tag{37}$$

Proof. Set $j = \mathbf{C}\overline{\beta}b$ and $h = \mathbf{C}\chi a$. Then, by (3) assumption (36) yields

$$\mathbf{C}\overline{\alpha}a = h \cdot j \ , \quad \text{for all} \quad a: X \to A \tag{38}$$

Set $r = S\sigma$. Then, by (1) we also have, for all such a's and $\sigma \in \Sigma$,

$$\mathbf{B}_r(\mathbf{C}\overline{\alpha}a) = \mathbf{B}_r h \cdot \mathbf{B}_r j \quad . \tag{39}$$

(Codomain) Since j is onto B, we can prove $(h \cdot \beta\sigma) \cdot \mathbf{B}_r j = (\alpha\sigma \cdot \mathbf{B}_r h) \cdot \mathbf{B}_r j$, for all α and σ as above (instead of $h \cdot \beta\sigma = \alpha\sigma \cdot \mathbf{B}_r h$). This follows from (10), i.e. from $\mathbf{C}\overline{\alpha}a \cdot \tau\sigma = \alpha\sigma \cdot \mathbf{B}_r(\mathbf{C}\overline{\alpha}a)$. In fact, the right hand sides are equal by (39), while (38) arranges the left hand sides, since by (10) $j \in \mathsf{H}\tau\beta$.

(One to one) A same h for two different a's cannot occur in (38), since $\mathbf{C}\overline{\alpha}$ is one to one by (10).

(Onto) Consider any $k \in \mathsf{H}\beta\alpha$. When $a = k \cdot b$, we get $k = h$ from **1.5**. In fact, $a = h \cdot b$ follows from (38) by the inverse of (10). *Q.D.E.*

5.1 Definition. A family $b: X \to A$ is called a *synthetic basis* of an algebra α on A if $\mathbf{r}b: \mathsf{H}\alpha\alpha \twoheadrightarrow \mathsf{F}XA$, where $\mathbf{r}$ is the CB–function of α and α at X, as in **1.2**.

5.2 Definitions. Keep the previous notation. Then, we call b an *analytic basis* of α if there exists a single $\chi: A \to \mathsf{F}(\mathsf{F}XA)A$ such that

$$\overline{\alpha} = \chi \cdot \mathbf{C}\overline{\alpha}b \quad , \tag{40}$$

viz. if we can denote any algebraic form in a single manner by the algebra element that is its value at b. Hence, χ is the only algebra with a constant X–valued species on A that represents $\overline{\alpha}$ and we call it the *analytic representative* of $\overline{\alpha}$ under the *reference system* b. If we drop the uniqueness of χ, then we call b a *(Marczewski) independent family* of α.

5.3 Lemma. *Let b be an analytic basis as above, then* $\mathbf{C}\chi\colon \mathsf{F}XA \rightarrowtail\!\!\twoheadrightarrow \mathsf{H}\chi\chi$ *and* $\mathbf{C}\chi b = \mathbf{i}_A$. (Notice that the statement $\mathbf{C}\chi a \in \mathsf{H}\chi\chi$ merely is (an extension of) "superassociativity" [27] and that our identity expresses one of the properties of a "selector system".)

Proof. α has to be connected with b. Otherwise, we could define more χ's satisfying (40). This gives us our identity. In fact, since each $a \in A$ is an $\overline{\alpha}tb$, by (40) $a = \chi ab = \mathbf{C}\chi ba$ for all a's.

To get our bijection we use **5.0** and prove that $\mathsf{H}\alpha\alpha = \mathsf{H}\chi\chi$. By (9) $\mathsf{H}\alpha\alpha \subseteq \mathsf{H}\overline{\alpha}\,\overline{\alpha}$ and by (40) $\mathsf{H}\overline{\alpha}\,\overline{\alpha} \subseteq \mathsf{H}\chi\chi$. In fact, $h\cdot\overline{\alpha}t = \overline{\alpha}t\cdot\mathbf{B}_X h$ becomes $h\cdot\chi a = \chi a\cdot\mathbf{B}_X h$ with $a = \overline{\alpha}tb$, while the required quantification follows from our connectedness. Hence, we only have to prove that $\mathsf{H}\chi\chi \subseteq \mathsf{H}\alpha\alpha$ or by **5.0** that $h \in \mathsf{H}\chi\chi$ implies $h = \mathbf{C}\chi(h\cdot b)$. This equality easily follows from the assumption $h\cdot\chi a = \chi a\cdot\mathbf{B}_X h$ (apply b to both sides, use our identity and abstract a by $\mathbf{C}$–reduction). *Q.D.E.*

5.4 Definitions. Given sets A and X and a function $\chi\colon A \to \mathsf{F}(\mathsf{F}XA)A$, consider the binary operation $\circ\colon \mathsf{F}XA\times\mathsf{F}XA \to \mathsf{F}XA$, defined in infix notation by by $a'\circ a'' = \mathbf{C}\chi a'\cdot a''$ for all $a', a''\colon X \to A$. This operation is called the *(restricted) matrix product* of an algebra α on A, if $\mathbf{C}\chi$ is an isomorphism from it onto the (converse composition of the) endomorphism monoid of α. The unit $b\colon X \to A$ of the matrix product is called a *geometric basis* of α.

5.5 Definitions. Consider the power $\mathcal{F}(\mathsf{F}XA)\alpha$ of an algebra α on A to the set of families $\mathsf{F}XA$, as in **3.0**. Let p denote the family of *projections* $\mathbf{Ci}_{\mathsf{F}XA}$. Then, the subalgebra ϕ of $\mathcal{F}(\mathsf{F}XA)\alpha$ connected with p is called the *conjugate space* of α and its carrier $\mathbf{c}p$ clearly is the set of the algebraic forms of α over X as in **1.2**. A family $b\colon X \to A$ is said to be a *GRN* (<u>G</u>oetz and <u>R</u>yll–<u>N</u>ardzewski [16]) *basis* of α, if there is an isomorphism $\chi\colon A \rightarrowtail\!\!\twoheadrightarrow \mathbf{c}p$ between α and ϕ, such that $\chi\cdot b = p$.

5.6 Theorems. *Our four definitions about bases are equivalent. Moreover, given a basis, both χ's in* **5.4** *and* **5.5** *are equal to the (single) analytic representative. Conversely, each of these functions determines a single basis.*

Proofs. (**5.1** implies **5.2**) Define $\chi\colon A \to \mathsf{F}(\mathsf{F}XA)A$ by $\chi = \mathbf{C}(\mathbf{r}b^{-1})$. Thus, by (2), for each $a\colon X \to A$, $\mathbf{C}\chi a$ is an endomorphism h such that

$$h\cdot b = a \quad . \tag{41}$$

Set $k = \mathbf{C}\overline{\alpha}a$ and $j = \mathbf{C}\overline{\alpha}b$. In order to get (40) we can show by (2) that $\mathbf{C}(\chi\cdot j)a = k$ for all a's. By (10) $k \in \mathsf{H}\tau\alpha$ and $\mathbf{C}(\chi\cdot j)a = h\cdot j \in \mathsf{H}\tau\alpha$ by (3), (10) and **1.1**(C). Hence, by the inverse of (10) the required equality $h\cdot j = k$ comes from (41).

By (40) we can prove that χ is single by proving that α is connected with b. Let β in **5.0** be the subalgebra connected with b. Since by (2) $\mathbf{C}\chi\cdot\mathbf{r}b = \mathbf{i}_{\mathsf{H}\alpha\alpha}$, (37) implies $\mathsf{H}\alpha\alpha = \mathsf{H}\beta\alpha$, which always implies $A = B = \mathbf{c}b$.

(**5.2** implies **5.4**) Keep χ and observe that by **5.0** $\mathbf{C}\chi\colon \mathsf{F}XA \twoheadrightarrow \mathsf{H}\alpha\alpha$. The remaining homomorphic condition (and the required unit of the matrix product) follow from **5.3**. In fact, for all $a', a'' : X \to A$ and $a \in A$, $\mathbf{C}\chi a'(\chi a a'') = \chi a(\mathbf{C}\chi a' \cdot a'')$ implies $\mathbf{C}\chi a' \cdot \mathbf{C}\chi a'' = \mathbf{C}\chi(a' \circ a'')$ by $\mathbf{C}$–reduction and by abstraction of a.

(**5.4** implies **5.5**) Keep χ. We get the required family of projections from b being a right unit, $a \circ b = a$, by (3), (2), the abstraction of $a\colon X \to A$ and (2) again. Since it also is a left unit, $(b \circ a)x = ax$ for all $a : X \to A$ and $x \in X \neq \emptyset$, we easily get the required injectivity (which is trivial when X is empty) by (3), $\mathbf{C}$–reduction and $\mathbf{B}$–reduction. Then, we get the homomorphic condition $\chi(\alpha\sigma a)a' = (\alpha\sigma \cdot \mathbf{C}(\chi \cdot a))a'$, for all $a : S\sigma \to A$ and $a' : X \to A$, from the one for $\mathbf{C}\chi a' \in \mathsf{H}\alpha\alpha$ by $\mathbf{C}$–reduction, (3) and $\mathbf{B}$–reduction.

(**5.5** implies **5.1**) For all $a\colon X \to A$, $\mathbf{C}\chi a \in \mathsf{H}\alpha\alpha$. (Get it from the isomorphic condition by reversing the passages just outlined.) Moreover, b has to be a generator of α because of our isomorphism. Therefore, by **1.5** it suffices to prove the ontoness in **5.1** by proving that $\mathbf{r}b \cdot \mathbf{C}\chi = \mathbf{i}_{\mathsf{F}XA}$. This is a rewriting of $\chi \cdot b = p$. In fact, the latter equality becomes $\mathbf{C}(\chi \cdot b) = \mathbf{i}_{\mathsf{F}XA}$ by (2) and then $\mathbf{r}b(\mathbf{C}\chi a) = a$ by application and (3).

(Moreover) In the preceding four steps all the χ's and $\mathbf{r}b$ were related each other through bijective constructions. Hence, it remains to show that $\mathbf{r}$ is one to one, which is trivial. (Take $h = \mathbf{i}_A$ in its definition **1.2**.) *Q.D.E.*

5.7 Corollaries. Given $a\colon X \to A$, consider the associated endomorphism $h = \mathbf{C}\chi a$ with respect to b as above. Then, (A) *a is independent iff h is one to one;* (B) *a is a generator of α iff h is onto A;* (C) *if a is a basis for α, then its analytic representative is*

$$\chi' = \chi \cdot h^{-1} \ . \tag{42}$$

(D) *A set $E \subseteq \mathsf{F}AA$ is the set of all endomorphisms of some algebra on A having a base iff composition on E forms a monoid and there is a family $b\colon X \to A$, for some X, with the property that the function $\mathbf{r}b\colon E \to \mathsf{F}XA$, such that $\mathbf{r}bh = h \cdot b$ for all $h \in E$, is a bijection onto $\mathsf{F}XA$.*

Proofs. Let $j = \mathbf{C}\overline{\alpha}b$. (B) Since j is onto A, this follows easily from (38).

((A) and (C)) Since j is onto A, by (38) and (40) $\overline{\alpha} = \chi' \cdot \mathbf{C}\overline{\alpha}a$ is equivalent to $\chi = \chi' \cdot h$. Therefore, when a is independent, h is one to one, since by **5.5** χ is, and $\chi' \cdot \mathbf{i}_C = \chi \cdot h^{-1}$, where $C = h^{\uparrow}A$ equals A by (B), when a is a basis. Conversely, when h is one to one, we define χ' on C by (42), we extend it on A and use our equivalence to find a independent.

(D) (Only if) See **5.1**. (If) The function $\chi = \mathbf{C}(\mathbf{r}b^{-1})\colon A \to \mathsf{F}(\mathsf{F}XA)A$ is an algebra on A (with a constant species). Since $\mathbf{C}\chi\colon \mathsf{F}XA \twoheadrightarrow E$, we prove that $h \in \mathsf{H}\chi\chi$ iff $h = \mathbf{C}\chi a$ for some $a \in A$. When $h = \mathbf{C}\chi a$, the monoid assumption yields $\mathbf{r}b(h \cdot f) = h \cdot f \cdot b$ for all $f = \mathbf{C}\chi d \in E$. Apply both sides to $\mathbf{C}\chi$, then any $c \in A$ to them: you get $h(\chi cd) = \chi c(h \cdot f \cdot b)$ by $\mathbf{C}$–reduction. Then, get $h \cdot \chi c = \chi c \cdot \mathbf{B}_X h$ as required by abstracting $d = f \cdot b$. Conversely, if this homomorphic condition holds everywhere, then apply b to both its sides, observe that $\mathbf{C}\chi b = \mathbf{i}_A$ and by abstracting c get $h = \mathbf{C}\chi a$ with $a = h \cdot b$. *Q.D.E.*

5.8 Definitions. Keep the notation of **5.4** and let B be the set of bases of α with index X. Consider a family $k: B \to \mathsf{F}(\mathsf{F}XA)C$, for some C, and a $b \in B$. If for all $b' \in B$ and $M: X \to A$

$$kbM = kb'(M \circ b') \quad , \tag{43}$$

then we say that the function $kb: \mathsf{F}XA \to C$, as well as family k, are *(absolutely) invariant.* (In fact, if kb is invariant, then any kb' is, as it follows from **5.1** through easy passages.) See [38] for a (lone) definition of universal invariance in F. Klein's synthetic form.

We call an invariant function $\eta: \mathsf{F}XA \to D$ *general* if for any invariant family k as above there is a function $f: B \to \mathsf{F}DC$, such that $kb' = fb' \cdot \eta$ for all $b' \in B$.

5.9 Corollary. *If χ is an analytic representative, then $\eta = \mathbf{C}\chi$ is a general invariant function.*

Proof. By **5.6** $\eta = \mathbf{r}b^{-1}: \mathsf{F}XA \to \mathsf{H}\alpha\alpha$. Hence, η is an invariant function if (43) holds for the k such that $ka = \mathbf{r}a^{-1}$ for all $a \in B$, i.e. if $\mathbf{r}b'(\eta M) = M \circ b'$ for all $M : X \to A$ and $b' \in B$. This is a trivial identity because of our notation of $\mathbf{r}$, η and $\circ$ in **1.2**, **5.9** and **5.4**. Also, η is general, since it is one to one. (We can take $fa = ka \cdot \eta^{-1}$ for all $a \in B$.) *Q.D.E.*

5.10 Example. By the preceding corollary any function or predicate of the endomorphism associated to a family M is invariant. Hence, in based algebras all present theory about universal eigenvalue equations concerns invariants only. E.g. by **5.7**(A) independence is an invariant predicate. This differs from Universal Algebra, where noninvariant notions are accepted, as we are going to show. In fact, we disprove the invariance of the "$\mathcal{C}/\mathcal{C}_i$–independences" as in [15]. These are weaker notions of independence, sometimes [11] and [17] considered akin of independence itself. In a vector space, they express the lack of certain linear dependencies among the elements (columns) of a family M and are equivalent to the independence of M. Here, we recall a couple of them that correspond to conditions (C_3) and (10) of [15]. (However, it is easy to see that the next conterexample works even for conditions (C_1) and (C_2) ibidem.)

Given M and α as in **5.8**, consider the following two conditions:

$$\bigcap(C^{\uparrow}V) = C(\bigcap V) \ , \quad \text{for all } V \subseteq \mathsf{P}M \tag{44}$$

and

$$Mx \in Cv \quad \text{implies} \quad Mx = vx \ , \quad \text{for all } v \subseteq M \text{ and } x \in X \ , \tag{45}$$

where $C\colon PM \to PA$ is such that $Cv = \mathbf{c}v$, when v is not empty, otherwise it denotes the set of the *algebraic constants* (see conditions (T) and (G) in [26]). As shown in [26] up to a minor restriction, the independence of M implies both conditions, whereas the converses do not hold. The very counterexample, used there to disprove the implication from (44) to independence, can be modified in order to disprove the invariance of both conditions.

Let α consists of a single operation $+$, defined as the direct square of the sum $\bmod 6$. Let $X = \{x, y\}$ and consider the four families over it listed in the table below. Check (40) and get χ for the reference system b. ($\chi\langle i, j\rangle$ is the polynomial function of $ix+jy \bmod 6$.) Then, check that b' too is a basis by **5.7** and that $M' = M \circ b'$.

X	b	b'	M	M'
x	$\langle 0,1\rangle$	$\langle 1,2\rangle$	$\langle 0,2\rangle$	$\langle 0,2\rangle$
y	$\langle 1,0\rangle$	$\langle 1,3\rangle$	$\langle 0,3\rangle$	$\langle 0,5\rangle$

Now, M satisfies (44) as well as (45), whereas M' does not satisfy either one. Thus, (43) fails in both cases, i.e. when kb expresses either (44) or (45). (Conversely, if we like the negations of (44) or (45), then the reverse changement of the reference system yields the noninvariance of the failures of these conditions.) Hence, both conditions are not invariant.

(This merely means that these conditions are not relevant to our universal algebras. They still might be relevant to a more specialized "geometry" of some interest on these algebras, in the same way as e.g. distance, a notion not relevant to vector spaces, is relevant to their metric geometry. However, from the extensive survey of [15] one finds that such a problem has never been considered for these conditions.)

5.11 Comment. While Marczewski's independence is a global condition for a family M, the conditions previously considered merely involve some "pieces" of M (e.g., for a matrix M, its columns). Hence, the latter conditions *break* independence, whenever they are equivalent to it, as in vector spaces. This breaking ability was an earliest motivation for studying them as in [37] and [25]. However, after Marczewski's proof that these conditions are weaker than independence (and his comments at p. 45 and 46 of [26]), the problem of breaking independence was relinquished. It was not included in the lists of the open problems of Colloquium Mathematicum nor of G. Grätzer. It also is not mentioned in [15].

At now, we know that independence is breakable at least in a broad sense. In fact, by **5.7**(A) we have necessary and sufficient conditions for it as in **1.9**(A), which involve parts of the endomorphism associated with M. Besides, since the previous conditions are not invariant, while independence is, to disprove their equivalence with independence is a poor evidence for the lack of breaking characterizations. Hence, the problem of finding (further) characterizations of independence could deserve some more attention.

6 Combinator aided Algebra (appendix).

6.0 The trees of Flatland. Our Fr–operations of **1.0** are "set-ary", viz. the arguments of an operation are indexed by any set. On the contrary, conventional operations of Universal Algebra are either n–ary or α–ary [17], viz. their arguments either are structured as n–tuples or are indexed by a set having some structure (an ordinal). In both cases, there is an order that carries over to related objects such as polynomial trees (ordered sons of a node) or their frontiers (ordered families of leaves of a tree and strings of their labels).

Our preceding treatment has shown that those order structures are not needed at all. Here, we exhibit an example where they also conflict against the natural structure of an operation occurring in an application. (Still, we have chosen set–arity mainly for technical reasons, as the simplified treatment of section **1** or the extensions in **6.9** and **6.11**.)

Let us consider a syntactic compression code [33] such as the Cascade Division Coding [43]. This is a representation of planar black and white pictures of a high resolution. Any picture is represented by a tree, where the leaves are binary labeled and the other nodes are singleton labeled. It is an efficient encoding, when the subset of the black pixels is small, as it often happens after a contour extraction.

If we embed a picture into a square grid of pixels with an edge being a power of two as in fig. 1, then we get its representing tree as in fig. 2 by a procedure consisting of an (isomorphic) tree of "divisions". At each division we cut a square into four subsquares and we mark the completely white subsquares by 0 and the others by 1. Any 1 marked subsquare becomes the square for a successive division, until we reach the desired resolution.

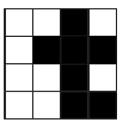

Fig.1

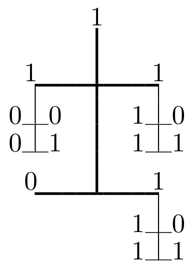

Fig. 2

Of course, we can formalize such trees as the trees of some conventional symbolic polynomials. Yet, a conventional order on the descendants of a node is as meaningful as a Peano's curve filling a square! In fact, we get such a fuzzy curve as the conventional frontier of "the" tree, which represents a square of black infinitesimal pixels.

6.1 Set-ary operations. The above mentioned mismatch between the conventional formal structure and the real life one occurs in other cases as well. In theoretical biology [36], we have to introduce ad hoc tricks (even in the simple example of the red algæ) in order to match the two structures. The conventional order is doubtful even in the earliest application: Formal Language Theory for programming languages. In fact, the parser of a real compiler works as if the programming language be context free in spite of the conventional theory, which says the opposite. However, if we are allowed to run a theoretical model, e.g. the one proposed in **3.9**(C), as the compiler (or even the programmer) does, then some theoretical cases against context freedom, such as a declarative instruction, vanish. This entails that a program be a data structure more complex than a string. Again, we have frontiers with a structure that does not match the one derived from the conventional order.

These examples are instances in a more general problem. Since a set can be well ordered, a (pure) set does not receive any harm, when the conventional order is added to it. Unfortunately, pure sets are not common in the real world. Sets often have their own structure, which can conflict against the added one (that merely represents our habit of left to right writing). Such a conflict may not arise for the logician, who handles left to right written objects, still it is dangerous elsewhere e.g. for computational purposes. It helps to conceal structural features of computational interest such as parallelism. Therefore, a pure set is an index safer than an ordinal for universal purposes. Anyhow, if the linear order fits some application, we always can choose an ordinal as index set within our formalism.

Sets in lieu of ranks just are one improvement for the universal notion of operation. Others are needed in order to present operations and algebras for specific applications with more flexibility. In such a wider environment the very combinatory machinery, we introduced for set-ary operations, works again up to minor changes, as we will show in **6.8**–**6.10** for some other presentations of operations, whereas from conventional Universal Algebra one has to climb a level of abstraction, as done in [24], and to lose intensionality.

6.2 Lemata. (A) *Given any nonempty function S, there are not functions τ, defined everywhere, with values $\tau_X \in \prod_{\sigma\in\Sigma} \mathsf{F}(\mathsf{F}S_\sigma T_X)T_X$, where Σ denotes the domain of S and T is any function defined everywhere, such that for all X*

$$X \subseteq T_X \ , \tag{46}$$

$$Y \subseteq X \quad \textit{implies} \quad \tau_Y\sigma \subseteq \tau_X\sigma \quad \textit{for all } \sigma \in \Sigma \quad \textit{and} \tag{47}$$

$$\tau_X\sigma t \neq x \qquad \textit{for all } \sigma \in \Sigma,\ t\colon S\sigma \to T_X \textit{ and } x \in X \ . \tag{48}$$

(B) *Furthermore, there are not such* τ*'s even when their domain (and the corresponding quantifications) are restricted to all* X *disjoint from* Σ*, unless for all such* X

$$DT_X \subseteq \Sigma \quad , \tag{49}$$

where $DT_X = \{\tau_X \sigma t \mid \sigma \in \Sigma \text{ and } t\colon S\sigma \to T_X\}$.

Proofs. (A) By contradiction, assume τ satisfies (46) and (47). Since $S \neq \emptyset$, there exists $\sigma \in \Sigma$. Let $Y = S\sigma$, take any $t\colon S\sigma \to Y$ and set $d = \tau_Y \sigma t$. Then, when $X = Y \cup \{d\}$, (48) fails because of (46) and (47).

(B) By contradiction, assume there is Y disjoint from Σ, such that $D' = DT_Y - \Sigma \neq \emptyset$. Then, take any $d \in D'$ and set X as in (A). Clearly, $X \cap \Sigma = \emptyset$ and (48) fails again. *Q.D.E.*

6.3 Conventional term definitions. Our assumption in **1.2** that composed terms and unknowns are disjoint is close to the similar assumption about combinatory terms [20]. The preceding statement shows that it is essential in a universal setting, where we would like to choose the unknowns freely. In fact, as one cannot have a natural number successor without the first Peano's axiom, so one cannot have anarchic algebras without (48), the first Peano axiom. It also concerns real applications. E.g., when we write a parser for a syntax directed translator, care has to be taken in order to avoid that a terminal could be taken as a nonterminal (or as a rule in the approach of [41]).

Condition (46) provides us the simplest unambiguous set-theoretical injection between unknowns and terms. The resulting construction of terms is very simple and set-theoretically natural. Hence, from an intensional point of view it is worth the price of the disjointness restriction in **1.2**. Besides, in conventional free algebras, the free generators *are included* in the generated algebra and cannot be freely chosen.

On the contrary, if we disregard intensional simplicity, then we can choose the unknowns freely. Indeed, in [24] another construction of terms gives up (46) and always gets anarchic algebras with terms full of delimiters and labels, in order to define its "theory functor" on all the category of sets. E.g., the generator corresponding to x becomes the term $\langle 0, \langle 0, x\rangle\rangle$, see **1.1.7**, **1.1.11** *and* exercise 2 of **1.1** ibidem. (However, its notational convention in **1.1.7**, which conceals the failure of our (46), yields a notational ambiguity later on. E.g., it is false that its "total description map" is an extension of an identity, as it appears in **1.1.18** ibidem.)

Anyway, both constructions do not violate **6.2**. On the contrary, the conventional constructions of Universal Algebra ([17], [7], [23], [29] and [4]) do. In fact, in spite of some restrictions about the unknowns and the alphabet, these accept (46) and (47) and easily allow us to construct a τ as in **6.2** either (A) or (B). (Notice also that, if one could accept the proviso in (49), one should render Σ a proper class, but for trivial cases.) Hence, the conventional treatments of polynomials are inconsistent.

(A possible explanation of this inconsistency is that $X \cap \Sigma = \emptyset$ concealed the need of our first Peano axiom. Perhaps, that useless condition is related with taking a catenation monoid, considered below, as a primary notion.)

6.4 Implementing terms. Another departure of **1.2** from the conventional constructions and from the one of [24] is that we merely use the simplest set-theoretical notions (functions and pairing) to build terms. On the contrary, terms are elsewhere defined as strings with, or better without, parentheses. Set-theoretically this is cumbersome, since it requires the more complex notions of words and of catenation, perhaps together with some ad hoc restrictions as in [24]. Yet, from a categorical point of view, things reverse, since a catenation monoid becomes a quite natural notion. Hence, let us see how things look from a non theoretical point of view.

The set-theoretically simpler terms are the fitter ones for implementation purposes on present computers. In fact, a pair $\langle \sigma, t \rangle$ is immediately implemented by a double word with two pointers, e.g. the former points to σ in a species or phylum table and the latter to an array t of pointers for a direct access file of terms. This kind of linked data structure is well known for its ease both in computation and in memory management.

On the contrary, strings require stacks as additional devices to perform most of the computations on terms (which require to parse their trees). Moreover, to represent the very string defining a term is a problem similar to the one of representing a term. At least for memory management purposes we would resort to a linked data structure again.

6.5 Mispairings. In **1.0** we also avoided to consider an algebra as a pair of a carrier and of a set or a family of operations. Such a pairing wastes ink, prevents the extension of carriers to proper classes (see **1.31** in [28]), conflicts with set-theoretical practice and is a possible instigator of some recognized misdeeds as we are going to recall.

In pure Set Theory, functions are not decoupled from their domains, which indeed are derived from them, and we saves both notation and theory as in [28]. Such savings become more worthy in case of algebras, which already are more complex than functions. Besides, a pair is a certain set-theoretical construction with its own properties. To use it just to say "A is the carrier of α" seems a misuse. E.g. this is not how Combinatory Logic consider the interplay between types and combinators. There, we introduce specific notions for this and one does not use pairing objects such as "dyads", in spite of a better chance to do it. (Combinatory terms can be pure intensional objects, independent from their types, contrary to the objects of Set Theory.)

In Algorithmic Information Theory [5], a close counterpart of the notions of domain and carrier is the one about the location of an input program, expressed by its length. (Such a location becomes a domain identification in case the input program consists of a finite table.) In the early developments (not in the present ones), input programs and their input lengths were decoupled as conventional Algebra does for algebras and their carriers. The outcome from this is expressed in [6]: “This seemingly minor point ... paralyzed progress in the field for nearly a decade ...”.

One still could be optimistic about Universal Algebra being more lucky. Yet, this misuse of pairs has spread elsewhere. In more applied fields, such as Theoretical Computer Science, triples, quadruples and so on abound. A side effect of this is the possible confusion of the different logical roles of the elements of such n–tuples. E.g. such a confusion is the one between constant operations and unknowns in (finite) algebras that was responsible of the delay between the partial solution [42] and the complete one [41] of the problem of characterizing the regularity of derivation trees. (Recall that often in such n–tuples we find “initial states” that logically are assignments for unknowns.)

6.6 Partial functions and the successor–set theory. Conventional Universal Algebra lacks the notion of CB–function in **1.2** and the related results of section **5**, possibly because it underplayed the “matrices” M. We exhibit a further example supporting the claim of **0.2** that the real objects we encounter in the applications are families M representing endomorphisms h, instead of the endomorphisms themselves. In fact, all the cases of M’s, so far considered, share the fact of being much smaller than the corresponding h’s. Hence, one might guess that some low practical reason has led people to ignore the h’s. On the contrary, here we find an M and an h that are almost indistinguishable by size.

Consider the empty valued singleton species $S\colon 1 \to 1$, viz. the class of the algebras with a single constant operation. Clearly, any two algebras with the same number of elements are isomorphic and no carrier can be empty. Hence, we do not lose generality when, given any set X, we choose the algebra α on the successor set of X, $A = X \cup \{X\}$, such that $\alpha\emptyset\emptyset = X$.

As known (see exercise 5 in **1.1** of [24]), the endomorphisms of an algebra of species S can be thought of as the partial functions on a set. For our α, this is to say that any endomorphism identifies a partial function on X and conversely. In fact, such an endomorphism is given by a “table” on X, filled either by the elements of X or by X itself. Such an X is a “don’t care” entry, e.g. a blank. (We do not need the whole “table” of the endomorphism on A, since its entry at X is constantly set at X by the endomorphism condition.)

More formally, $\mathbf{i}_X\colon X \to A$ is an associating family as in **1.2** and a base b as in **5**. Any $M\colon X \to A$ corresponds to an $h = \mathbf{C}\chi M\colon A \to A$, where $h = M \cup \{\langle X, X\rangle\}$. Therefore, h and M are almost the same table. Yet, in practice nobody adds the lone row “don‘t care goes into don‘t care” to a partial table M.

(In a vector space, the computation relevant object clearly is a matrix and not its linear transformation. Yet, the latter one might also appear as a real life object to a mathematician of today. However, some historical evidence hints it does not at a mathematically unbiased eye. The perspective experiment of F. Brunelleschi, the earliest precise "definition" of a linear transformation, indeed, consisted of an analogically computed matrix (the painting), corresponding to a beautiful (redundant) reference system (the baptistery), drawn out from the space (the sky and surroundings, mirrored twice).)

6.7 Cospectral recognition. In a vector space the eigenvalues can be used to draw out a small invariant information from a larger and noninvariant one. Such a reduced information serves to recognize certain interesting features of a *natural* object by two ways that we might call *spectral* and *cospectral*. An example of the spectral way is color vision, where we use the light spectrum (through a frequency smeared sampling). An example of the cospectral way is a phenomenon of acoustic recognition, described in [44], which concerns us. Within cospectral recognition we disregard the amplitudes of a Fourier series and sees or "hears" the eigenvalues (frequencies) only. In the experiments reported in [44] people were able to perceive a *missing* fundamental frequency.

Can we find experimental instances of cospectral recognition also outside vector spaces? In general, when we recognize some feature of a class of objects, we merely gather some of these objects (the ones sharing the feature). In our case, the objects are our M's, viz. heterogeneous objects. The features are given by either their innermorphisms or their eigenmorphisms. In case of an affirmative outcome, an experiment should prove e.g. that humans or other test subjects are able to gather the (presentation of the) M's by their cospectra as in **2.1**.

(On the contrary, spectral recognition uses the eigenvalues (wavelengths) only indirectly, since its relevant quantities are the amplitudes (at the wavelengths). Though it is interesting, we are not yet equipped to make it universal. In fact, we should have a treatment of *universal transforms* able to universalize vector space notions as the Fourier transform. Such a treatment might spring from the one for universal integration of section **2**. Anyway, it should concern the various (discrete time) "behaviors" v of a given M as in (16), not our objects M.)

Though experimental, this problem gives rise to a few mathematical questions, which now receive affirmative hints. The first one is: are we able to compute the cospectra of the objects chosen? There already is a program for it in the finite inner case [34]. Another question is: can we find non trivial cospectra for objects, simple enough to ease the experimenter, who has to present them to the test subjects? In **2.1** we found it for clocks with two bits only. A third requirement is that the objects show some nontrivial (viz. up to isomorphism) cospectrality. (It allows the experimenter to search for recognition mismatches due to cospectrality.) Again, we found it in **2.1**(D).

6.8 Aylotypes and rheomorphisms. In **3.7**, f is a rheomorphism from u exactly when the relation $\mathbf{e}fu$ is a function. Then, both u and $\mathbf{e}fu$ belong to sets of the type $\mathsf{F}x(\mathsf{F}Tx)$ for some set x. Hence, we can say that f is a rheomorphism exactly when it conveys information *flowing* within a *channel* of a given *shape*, which in our case is "$\mathsf{F}x(\mathsf{F}Tx)$ *up to* x", or, in a combinatory notation [20], our aylotype $\mathbf{S}\mathsf{F}(\mathsf{F}T)$.

In **3.7** we introduced the aylotypes and the rheomorphisms only in order to fulfill our requirements of **0.4** about the category U of the retro-operations as in **3.8**(E), (F), (I), (K) and (L) in a simple way. (An indirect treatment through the isomorphic category U' of the C–H homomorphisms was slightly more clumsy. E.g. u in (30), its "subject", would have been replaced by a $\mathbf{C}u$, if we had to use a $u \in U'$. Such a treatment was used in section **1**, see e.g. (10), only in order to keep it as close as possible to the conventional treatments.) However, in **6.9** we will find that these new notions have a larger extent. Hence, let us focus on how we got the extension $\mathbf{e}f$.

Our flowing information springs as a set-theoretical function $g = \{\langle a, b\rangle \mid b = ga,\ a \in A\}$ and sinks again as a set of pairs $\{\langle pa, qb\rangle \mid \langle a, b\rangle \in g\}$ (that has to be a function when (27) is satisfied). In this "parallel" mapping, both p and q can be determined by f through combinators: $p = \mathbf{p}f$ and $q = \mathbf{q}f$, where $\mathbf{p} = \mathbf{i}_{\mathsf{F}AB}$ and $\mathbf{q} = \mathbf{B}_T$, since $pa = fa$ and $qb = f \cdot b$ everywhere.

It is useful to relate the combinators $\mathbf{p}$ and $\mathbf{q}$ with our aylotype. This aylotype has the form "$\mathsf{F}(Yx)(Zx)$ *up to* x", when we set $Y = \mathbf{i}_X$, with an $X \ni x$, and $Z = \mathsf{F}T$. Then, we will say that these Y and Z are the *terms of* the aylotype and *correspond* to $\mathbf{p}$ and $\mathbf{q}$ respectively.

6.9 Intensional morphisms. Category theoreticians are allowed to disregard what precisely their "morphisms" are and can handle them with much generality. We, on the contrary, want to know how they are constructed. Yet, we need not to give up some generality. The aylotype–extension procedure, considered above, can be easily made into a general tool for constructing morphisms as the following examples will show.

(A) Let us change our aylotype into $\mathbf{W}(\mathsf{F} \cdot \mathsf{F}r)$, which is a combinatory notation for "$\mathsf{F}(\mathsf{F}rx)x$ *up to* x". (We can say that we only exchanged the terms of the old aylotype in **6.8**.) Then it is easy to check that the corresponding new "rheomorphisms", coming from a "$\mathbf{W}(\mathsf{F} \cdot \mathsf{F}r)$–extension" and the functionality requirement of **6.8**, are our ordinary homomorphisms from $\mathsf{F}r$–operations. (In fact, (28) changes into (4), after we have set $\mathbf{e}fg = \{\langle \mathbf{B}_r fa, f(ga)\rangle \mid a \in A\}$ for $f\colon A \to B$ and $g\colon \mathsf{F}rA \to A$.) Moreover, the categorical axioms about morphism composition now derive from simple set inclusion properties as in **3.8**(E) and (F).

(B) Now, take $\mathbf{W}(\mathsf{F} \cdot \mathsf{P})$ as our aylotype, viz. take "$\mathsf{F}(\mathsf{P}x)x$ *up to* x" as our channel shape. Again, our extension and functionality procedure provides us new rheomorphisms, which easily turn out to be the homomorphisms from the P–operations of [32]. (Set $\mathbf{p} = {}^{\uparrow}$

and $\mathbf{q} = \mathbf{i}_{FAB}$, then (28) changes into (6) of ibidem.) Also, it would be easy to check that this procedure works even for more exotic homomorphisms as the ones of **2.9** and **4.9** of [32].

(C) Furthermore, consider a *star operation* on A, defined as a function $f\colon {}^{\star}\!A \to A$, where ${}^{\star}\!A = \bigcup_{n \in N} FnA$. Star operations are not exotic at all. E.g., when $A = I^*$ as in **2.7**(A), the *general catenation*, i.e. the catenation of a variable number of words (corresponding to the clone of the catenation monoid), is such an operation $f\colon {}^{\star}(I^*) \to I^*$. Also, to vary the argument length of an operation might be useful, e.g. for formalizing length related notions of Formal Language theory as context–free determinism (through $LR(k)$ grammars) and the relation between deterministic push down automata and the corresponding tree acceptors.

Star operations have their category of *star homomorphisms*. In fact, given another star operation g on B and an $h\colon A \to B$, let $\mathbf{B}h\colon {}^{\star}\!A \to {}^{\star}\!B$ be everywhere defined by $\mathbf{B}$–reduction, $\mathbf{B}hwn = h(wn)$. Then, we can easily see that condition $h \cdot f = g \cdot \mathbf{B}h$ replaces (4) for finding such a category. Again, our homomorphisms are rheomorphisms. In fact, our aylotype is $\mathbf{W}(F \cdot {}^{\star})$, while $\mathbf{p} = \mathbf{B}$ and $\mathbf{q} = \mathbf{i}_{FAB}$.

(D) So far, we got homomorphisms for operations (perhaps exotic ones), viz. the arrow diagrams for the equations that set-theoretically characterize such rheomorphisms as (4) or (28) always are squares. But, there also are morphisms between objects that (usually) are not considered similar to operations and do not use squares.

Let us take the partition functions on a set r as our objects together with their order, defined in **0.7**, and let us see how to get their morphisms as rheomorphisms (as usual, we say there is a morphism from u to v iff $u \le v$). Since the equation in **0.7** characterizing such a morphism has a triangular arrow diagram, we have to make triangles into squares.

Combinatory logicians use a "constant generating" combinator $\mathbf{K}$ in order to have a function $\mathbf{K}c$ that, after application with any argument a, yields an arbitrary constant c, i.e. $\mathbf{K}ca = c$. This will provide us the missing angle of our diagram. In fact, consider the new aylotype $\mathbf{W}(F \cdot \mathbf{K}r)$, which is *a* combinatory notation for all the sets of the shape "$Frx = F(\mathbf{K}rx)(\mathbf{i}_X x)$ *up to* set x". Given an $f\colon A \to B$, we define its $\mathbf{W}(F{\cdot}\mathbf{K}r)$*–extension* as the function $\mathbf{e}f\colon FrA \to P(r \times B)$, such that $\mathbf{e}fg = \{\langle s, f(gs)\rangle \mid s \in r\}$ for all $g : r \to A$. In other words, $\mathbf{q} = \mathbf{i}_{FAB}$, since the corresponding aylotype term still is $Z = \mathbf{i}_X$, whereas $\mathbf{p} = \mathbf{K}\mathbf{i}_r$ that corresponds to $Y = \mathbf{K}r$. Then, given the partition functions $u\colon r \twoheadrightarrow A$ and $v\colon r \twoheadrightarrow B$, (27) holds iff $u \le v$. Again, the morphisms are rheomorphisms.

(We always start with the aylotype terms and then we get $\mathbf{p}$ and $\mathbf{q}$. Logically, one should do the opposite, since p and q are "combinatory terms" that determine their "principal types", which correspond to our aylotype terms (see **13.16** or **15.26** in [20]). Often, however, our way works better. In fact, this is quite similar to structured programming, where we write the executive instructions after the declaratives ones.)

6.10 Species and operator domains. In **1.0** we used a species S instead of an *operator domain map* Ω, which could be defined as the converse image map of S. In fact, for all the purposes, we faced so far, the simpler S was enough. However, an Ω, instead of an S was chosen in [24]. This choice was motivated in **1.1.4** ibidem by categorical reasons.

An Ω might be preferred to an S from a combinatory point of view too, once our purposes are extended. In fact, consider an algebra $\alpha \in \prod_{\sigma\in\Sigma} \mathsf{F}(\mathsf{F}S_\sigma A)A$. When S is a constant with value r, we can consider $u = \mathbf{C}\alpha$ as an operation of aylotype "$\mathsf{F}(\mathsf{F}rx)(\mathsf{F}\Sigma x)$ up to x", viz. we can handle an algebra as a single operation. Hence, we would like to do the same even when S is not constant, provided that we extend our aylotype notion. This is quite easy, if we use Ω. In fact, the new aylotype becomes "$\prod_{r\in\mathcal{R}} \mathsf{F}(\mathsf{F}rx)(\mathsf{F}(\Omega r)x)$ up to x", $\mathcal{R}$ being a domain for Ω.

6.11 Set extensive combinators. So far, we have seen how combinators and their related notions direct Set Theory toward intensional constructions. Here, we stress some requirements this role poses to present Combinatory Logic notions. The minor one is the introduction of new combinators such as the direct image map $^\uparrow$. (See [32] for another similar combinator, $^\downarrow$, and for their mutual "reduction" rules.) More interesting ones concern the typing of well-known combinators.

In **0.8** we defined the typed combinator $\mathbf{B}_X$ for relations. (This served in **1.0** to define congruences.) However, it was not a truly relational $\mathbf{B}$, viz. a prefix notation for relational composition that should have the principal "type" $\mathsf{F}(RAB)(\mathsf{F}(RXA)(RXB))$, where now R stands for "relations from ... to ...". It was a *semirelational* one, viz. a notation for certain *functionality* features of (nonfunctional) relations. Hence, we can consider it as a true combinator useful for our relative intensionality purposes.

These functionality features are disturbing: there is not a single way to express them, as it was in the case of the conventional (and fully functional) $\mathbf{B}$. In fact, in **0.8** our semirelational $\mathbf{B}_X$ was given two different (principal) types: $\mathsf{F}(RAB)(R(\mathsf{F}XA)(\mathsf{F}XB))$ and $\mathsf{F}(RAB)(\mathsf{F}(\mathsf{F}XA)(RXB))$. We have to motivate the intrusion of the latter odd looking $\mathbf{B}_X$ that occurred in **1.0** while defining congruences for $\mathsf{F}r$–operations. In fact, the former $\mathbf{B}_X$ suffices there, since the premise of the implication defining congruences could be rewritten as $\langle a, b\rangle \in \mathbf{B}_r e$ using the former $\mathbf{B}_X$.

The motivation for our type splitting comes from the ability of a premise of the form $\mathbf{G}ea = \mathbf{G}eb$ (as in **1.0**) to accommodate other cases of operations. In fact, for all operations of **6.9** we can set $\mathbf{G} = \mathbf{p}$ and get the corresponding congruences uniformly defined, whereas we cannot do it so simply by the other form of the premise. (The congruences for aylotype $\mathbf{W}(\mathsf{F}\cdot\mathbf{K}r)$ in **6.9**(D) are trivial, since any equivalence satisfies such a congruence implication, yet they exist.) Therefore, if we want such a uniform definition of congruences, then we have to split the type of semirelational $\mathbf{B}_X$.

Moreover, it is not very clear where the information about this splitting comes from. In fact, both $\mathbf{B}_X$'s of **0.8** are applied to the same e, while the further a again is the same. Hence, a possible extension of combinatory typing to this kind of types and combinators should also answer the problem of this *anomalous* type splitting.

(The combinator minded reader can also notice another type splitting in the case of the standard (i.e. functionally pure) $\mathbf{B}$. The (single) functional $\mathbf{B}_X$ in **0.8** is close to a *typed combinator* as in chapter 13 of [20], while $\mathbf{B}$ in **6.9**(C) is close to a *stratified* one as in chapter 14 ibidem. They, coexist even when we consider star operations only, because we need $\mathbf{B}_X$ for their X–anarchic extensions, which are defined as in **1.2** with (7) and (8) using a stratified version of our typed $\mathbf{C} = \mathbf{C}_B$ of **0.8**. However, after two applications we have the information about the right type. Hence, we can unify them as in chapter 16 ibidem.)

Two other problems come from the examples from **6.8** to **6.10** about rheomorphisms. The former is to formally define a general rheomorphism notion within Set Theory and to compare its extent with the one of the morphisms of categorical origin. The latter is to introduce rheomorphisms within Combinatory Logic by "free" definitions. (The definitions in our examples and **3.7** are not free: R is defined after F, yet F can be redefined by R through H from the homomorphisms of identity algebras.)

6.12 Combinators and Algebra. (A) A reason to consider our objects M as set-theoretical functions (as in **0.2**) is to satisfy certain modelling needs, preliminary to the choice of an algebra. (In fact, once the algebra is chosen, A is given and we might also think of our objects as morphisms $\langle X, M, A\rangle$ of the category of sets. Unfortunately, this might disregard the modelling step of the algebra choice. Sometimes, it is an "intuitive abstraction" step, i.e. it goes from some M's, found in an application, to some A's. One or few M's appear before the A's and multiple choices of an A can coexist. E.g., if a specific set M of D0L productions lacks empty strings, you might well take A as the set of nonempty words of a catenation semigroup.) Let us see an example where even the very Set Theory might be relinquished.

Consider D0L systems again and aim to a couple of "similar" sets of productions. The former, say $a \to ab$ and $b \to bb$, has a doubleton alphabet, while the latter, say $a \to ab$, $b \to bb$ and $c \to c$, has a tripleton alphabet. Set-theoretically, they are different sets of productions $M\colon X \to A$, with different X's. For a while let us fancy that they have to model some real life object, say certain "gray algae".

To a biologist, the latter set models gray algae, when he wants to see them attached to an inert foot c, and e.g. he use it with the seed ca, whereas the former models gray algae disregarding the foot, e.g. with the seed a. Yet, to him the gray algae growth is a single real life process and hardly his different modelling needs can split it. Therefore, with set-theoretical functions, a single set of productions (as well as its associated endomorphism) does not model his notion of gray algae growth very well. (Even the theory functor of catenation monoids could not work here, as the growth process is fixed.)

Our biologist could be more comfortable with multiple domain "functions" (and with distinct rules for singling out a domain). Possibly, such "functions" can be built within Set Theory. Yet, Combinatory Logic might be another solution. Pure combinators do not have (fixed) domains.

(B) Even within Set Theory, a surreptitious use of combinators offers various advantages. One is practical. Consider the bijective type assignments we encountered here. (See (10) together with **3.1**(B), (21), (30), (37) and the ones in **5.3** and **5.6**.) Their types represent categorical properties that, albeit theoretically interesting, are of little help, when we need to get effective results. On the contrary, the function to be assigned (the subject of the assignment) provides us such a practical information. E.g., the engineer of **2.0** uses the machinery of Switching Theory, i.e. the one (concisely) represented by the subjects in the bijective type assignments of section **5** (and often he ignores the corresponding universality property of the freedom of free finite Boolean set algebras).

Such practical advantages were already known in another environment as Programming Theory (see **3E** in [21] and **1.1** II in [2]). There, we interpret a type assignment of Combinatory Logic as a concise program (the term identifying its subject) together with its I/O specifications (the types). The programming tools coming from Structured Programming have been a successful test of this. The only change here is in the higher level of our "programming".

Combinators also have some heuristic value, at least in Author's experience at developing this theory. To develop a theory seems to be a trial and error procedure of heterogeneous steps. Some easy steps have a decidable nature and often correspond to the combinatory notions of abstraction, application and conversion. Here, the combinatory machinery is clearly useful. Generalizations are harder steps, yet not of a "magic intuition" nature. Combinatory Logic, in fact, tells us something.

E.g., when we want to universalize the definition d of form of higher degree of vector spaces into the definition Dx of a form of higher degree of a varying algebra x, we choose a property π that looks important in vector spaces (e.g. the Cayley–Hamilton theorem). Then, we search for a D such that Dx reduces to d, when x is a vector space v, and Dx satisfies π for all our x. A preliminary check in this search is to get the very Dv satisfying π. This is like an instance of "subject expansion" as in **14.26** of [21]. Hence, we can expect we have to downgrade π as hinted by **14.39** ibidem (and we did it), even disregarding other x's.

Sometimes, we go on through an even thicker fog, e.g. without perceiving the d's nor the π's. Then, we need some loud bell ringing, when an object or a property relevant to the theory is reached. Often, in Author's experience, this was the appearance of a well-known combinator, e.g. **C**, and/or of a conspicuous type, e.g. a *bijection*. (Combinator **C** occurs very often in the subjects of our type assignments. Yet, in part this is an artifice due to the present choice of algebra presentations. In [32], with other presentations, different "combinators" occur in some corresponding bijective type assignments.)

Once the heuristic phase is completed, again combinators serve to write down the formal proofs. In the present exposition, we partly obscured this technical role in order to be close to conventional treatments. Yet, a wider and unbiased use of combinatory notions could have shortened the proofs further (and improved the definitions). E.g. any combinator minded reader can improve the treatment of the forms of higher degree in **3.3** by replacing natural numbers with the iterators of Church.

(C) The notions used in the works of Universal Algebra are either "set-categorical" or categorical, i.e. other foundations are disregarded. In other fields, a "foundational monotheism" is not the present rule. Many recent works in Combinatory Logic (see [20] **10–12** and [2] **5** and **18–21**) use algebraic and categorical notions in order to study the combinators. E.g., we can characterize certain systems of typed combinators by cartesian closed categories [21]. It also was not the rule in the past. In classical (vector space) Mathematics, the intensional (analytical) approach consciously entwined the synthetic Geometry approach at least since eight centuries. (From the introduction of [13]: "Et que arismetrica et geometria scientia sunt connexe, et suffragatorie sibi ad invicem, non potest de numero plena tradi doctrina, nisi intersecantur geometrica quedam, *vel ad geometriam spectantia*, que hic tantum iuxta modum numeri operantur; . . .")

Acknowledgments. The earliest incentives for this work came from a stay in 1976 at the University of Waterloo (supported by the Canadian N.R.C.). Useful references came from J.R. Hindley and J.P. Seldin. The notion of star operation came from talks with Corrado Böhm. R. Giovanelli (University of Parma) provided the Author with equipment and advice on engineering topics. At the University of Milan, P. Laurence and G.C. Meloni improved the introduction, while A. Carboni, M. Bonecchi and M. Lunelli provided the Author with software assistance. B. Hayn pointed out a mistake in the original membership property definition. M. Hazewinkel and his referees of ACAP friendly hinted many improvements in style and presentation and a basic reference. The frank and pointed criticism of G. Ferrero (University of Parma) made the present exposition less unpalatable to algebraic readers.

References.

0 A.V. Aho and J.D. Ullman, *The Theory of Parsing, Translation and Compiling*, 2 vol.s, Prentice–Hall, Englewood Cliffs, 1972, 1973.

1 A.V. Aho, R. Sethi and J.D. Ullman, *Compilers: Principles, Techniques and Tools*, Addison-Wesley, Reading, 1986.

2 H.P. Barendregt, *The Lambda Calculus, its Syntax and Semantics*, 2^{nd} ed., North–Holland, Amsterdam, 1984.

3 C. Böhm, Tavola rotonda "Logica e Informatica: da incidenti di confine a un'alleanza d'interesse?", Intervento di Corrado Böhm, *Atti degli Incontri di Logica Matematica*, Dipartimento di Matematica, Università di Siena, **3**, X Incontro (1986), 349–357.

4 S. Burris and H.P. Sankappanavar, *A course in Universal Algebra*, Springer–Verlag, Berlin, 1981.

5 G.J. Chaitin, *Algorithmic Information Theory*, Cambridge University Press, Cambridge, 1987.

6 G.J. Chaitin, Randomness in Arithmetic, *Sci. Amer.*, **259**(1) (1988), 52–57.

7 P.M. Cohn, *Universal Algebra*, D. Reidel, Dordrecht, 1981.

8 K. Čulik II and J. Karhumäki, A new proof for the D0L sequence equivalence problem and its implications, in G. Rozenberg and A. Salomaa, *The Book of L*, Springer–Verlag, Berlin, 1986.

9 H.B. Curry and R. Feys, *Combinatory Logic, Vol. I*, North–Holland, Amsterdam, 1958.

10 H.B. Curry, J.R. Hindley and J.P. Seldin, *Combinatory Logic, Vol. II*, North–Holland, Amsterdam, 1972.

11 V. Dlab, Universal Algebra representation of regular GA–independence structures, *Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys.*, **17** (1969), 203–206.

12 H. Ehrig and B. Mahr, *Fundamentals of Algebraic Specifications 1: Equations and Initial Semantics*, EATCS Monographs on Theory Comp. Sci., Vol. 6, Springer–Verlag, Berlin, 1985.

13 L. Fibonacci, *Liber Abbaci*, 1202, reprinted by Tipografia delle Scienze Matematiche e Fisiche, B. Boncompagni ed., Rome, 1857.

14 E. Fried and J. Sichler, On automorphism groups of subalgebras of a universal algebra, *Acta Sci. Math.* **40** (1978), 265–270.

15 K. Głazek, Some old and new problems in the independence theory, *Coll. Math.* **42** (1979), 127–189.

16 A. Goetz and C. Ryll–Nardzewski, On bases of abstract algebras, *Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys.*, **8** (1960), 175–161.

17 G. Grätzer, *Universal Algebra*, 2^{nd} ed., Springer–Verlag, Berlin, 1979.

18 D. Gries, *Compiler Construction for digital computers*, John Wiley, New York, 1971.

19 J. Hartmanis and R.E. Stearns, *Algebraic Structure Theory of Sequential Machines*, Prentice-Hall, Englewood Cliffs (N.J.), 1966.

20 J.R. Hindley and J.P. Seldin, *Introduction to Combinators and λ–Calculus*, Cambridge University Press, Cambridge, 1986.

21 J. Lambek, Deductive systems and categories III. Cartesian closed categories, intuitionist propositional calculus and combinatory logic, *Lecture Notes in Mathematics* (Springer–Verlag) **274** (1972), 57–82.

22 W.A. Lampe, The independence of certain related structures of a universal algebra, *Algebra Universalis* **2** (1972), scattered pages.

23 H. Lausch and W. Nöbauer, *Algebra of Polynomials*, North–Holland, Amsterdam, 1973.

24 E.G. Manes. *Algebraic Theories*, Springer–Verlag, Berlin, 1976.

25 E. Marczewski, A general scheme of the notions of independence in mathematics, *Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys.* **6** (1958), 731–736.

26 E. Marczewski, Independence and homomorphisms in abstract algebras, *Fund. Math.* **50** (1961–62), 45–61.

27 K. Menger, Superassociative systems and logical functors, *Math. Annalen* **157** (1964), 278–295.

28 J.D. Monk, *Introduction to Set Theory*, McGraw–Hill, New York, 1969.

29 R.S. Pierce, *Introduction to the theory of abstract algebras*, Holt Rinehart and Winston, New York, 1968.

30 B.I. Plotkin, *Groups of automorphisms of algebraic systems*, Wolters–Noordhoff, 1972.

31 G. Ricci, Cascades of Tree–Automata and Computations in Universal Algebras, *Math. Systems Theory* **7**(3) (1973), 201–218.

32 G. Ricci, P–algebras and combinatory notation, *Riv. Mat. Univ. Parma* **5**(4) (1979), 577–589.

33 G. Ricci, Syntactic compression codes at the zero entropy point, *Intern. J. Computer Math.* **11** (1982), 221–226.

34 G. Ricci, Illogical programming, *Atti degli Incontri di Logica Matematica,* Dipartimento di Matematica, Università di Siena, **3**, X Incontro (1986), 417–420.

35 G. Ricci, Most definitions of trees over an alphabet are wrong, *Bulletin of the E.A.T.C.S.* **40** (1990), 265.

36 G. Rozemberg and A. Salomaa, *The mathematical theory of L–systems*, Academic Press, New York, 1980.

37 J. Schmidt, Mehrstufige Austauschstructuren, *Z. Math. Logik Grundlag. Math.* **2** (1956), 233–249.

38 J. Schmidt, Concerning algebraic independence (Abstract), *Coll. Math.* **14** (1966), 369–372.

39 B. Schweizer and A. Sklar, A grammar of functions I, *Aequationes Math.* **2** (1968), 62–85.

40 S. Shapiro ed., *Intensional Mathematics*, Studies in logic and the foundation of mathematics (North–Holland, Amsterdam) vol. 113, 1985.

41 M. Steinby, On algebras as tree automata, *Colloquia Mathematica Societatis János Bolyai, 17. Contributions to Universal Algebra*, Szeged (1975), 441–455.

42 J.W. Thatcher, Characterizing derivation trees of context–free grammars through a generalization of finite automata theory, *J. Comp. Syst. Sci.* **1** (1967), 317–322.

43 T. Usubuchi, K. Iinuma and T. Ishiguro, Data compression of facsimile signals by cascade division coding, *Conf. Rec. 1975 IEEE Int. Conf. Commun.* (1975), 7.22–7.25.

44 J. Walker, People listening to a bell can perceive sounds the bell does not really make, *Sci. Amer.* **251**(1) (1984), 118–123.