"The Arrival of the Fittest": Toward a Theory of Biological Organization

Walter Fontana[a] **and Leo W.Buss**[b]

[a]Santa Fe Institute
1660 Old Pecos Trail
Santa Fe, NM 87501 USA
(walter@sfi.santafe.edu)

[b]Department of Biology
and
Department of Geology and Geophysics
Yale University
New Haven, CT 06511 USA
(Leo_Buss@quickmail.cis.yale.edu)

Bulletin of Mathematical Biology

Contents

Abstract

The formal structure of evolutionary theory is based upon the dynamics of alleles, individuals and populations. As such, the theory must assume the prior existence of these entities. This existence problem was recognized nearly a century ago, when DeVries (1904) stated, "Natural selection may explain the survival of the fittest, but it cannot explain the arrival of the fittest." At the heart of the existence problem is determining how biological organizations arise in ontogeny and in phylogeny.

We develop a minimal theory of biological organization based on two abstractions from chemistry. The theory is formulated using λ-calculus, which provides a natural framework capturing (i) the constructive feature of chemistry, that the collision of molecules generates new molecules, and (ii) chemistry's diversity of equivalence classes, that many different reactants can yield the same stable product. We employ a well-stirred and constrained stochastic flow reactor to explore the generic behavior of large numbers of applicatively interacting λ-expressions. This constructive dynamical system generates fixed systems of transformation characterized by syntactical and functional invariances.

Organizations are recognized and defined by these syntactical and functional regularities. Objects retained within an organization are characterized by a grammar and interactions among objects by algebraic relationships. The objects which maintain themselves in the system realize an algebraic structure and possess a grammar which is invariant under the interaction between objects. An organization is self-maintaining, and is characterized by (i) boundaries established by the invariances, (ii) strong self-repair capabilities responsible for a robustness to perturbation, and (iii) a center, defined as the smallest set kinetically persistent and self-maintaining generator set of the algebra.

Imposition of different boundary conditions on the stochastic flow reactor generates different levels of organization, and a diversity of organizations within each level. Level 0 is defined by self-copying objects or simple ensembles of copying objects. Level 1 denotes a new object class, whose objects are self-maintaining organizations made of Level 0 objects, and Level 2 is defined by self-maintaining metaorganizations composed of Level 1 organizations.

These results invite analogy to the history of life, that is, to the progression from self-replication to self-maintaining procaryotic organizations to ultimately yield self-maintaining eucaryotic organizations. In our system self-maintaining organizations arise as a generic consequence of two features of chemistry, without appeal to natural selection. We hold these findings as calling for increased attention to the structural basis of biological order.

1 Introduction

The principal triumph of the modern synthetic theory of evolution was to recognize that alleles, individuals, and populations were entities relevant to evolution and that evolutionary changes in allele frequencies were governed by mutation, selection, and drift (Fisher, 1930; Wright, 1931; Haldane, 1932). From this realization came a rigorous, albeit loosely connected, body of theories that is widely acknowledged as collectively providing the formal basis of our understanding of evolution. Despite the undoubted success of population genetics and, by extension, population biology, this formulation of evolution suffers from an existence problem. A theory based on the dynamics of alleles, individuals, and populations must necessarily assume the prior existence of these entities. Present theory tacitly assumes the prior existence of the entities whose features it is meant to explain.

The existence problem may seem, at first glance, to be a problem limited to the origin of life. While it is true that the Darwinian process inexorably sets in once the relevant entities emerged, the problem of the origin of organizations nonetheless remains. This fact is reflected in the centrality of organizational issues in a multitude of unsolved problems. The genotype-phenotype mapping is a persistent problem in population genetics precisely because we lack fundamentally an understanding of how the phenotype is generated. Similarly, we lack an understanding of the limits on the origin of variation (e.g., developmental constraints), because we lack an understanding of how the individual emerges. Moreover, the origin of life is but the first in a progression of origins. The individual in the synthetic theory is not a single class (Buss, 1987). Multiple levels of organization have emerged in the history of life, and each such emergence raises the same existence problem as does the origin of life itself.

A clear conceptual distinction separates those problems solved within the framework of the Modern Synthesis and those which have proved recalcitrant. The traditional dynamics approach is appropriate when both the relevant entities (e.g., alleles, individuals and populations) and their kinetic couplings (e.g., mutation, selection, drift) are fixed and known. An alternative approach is required to understand how the entities themselves are generated. We have recently provided a brief report on a constructivist theory of biological organization grounded in λ-calculus (Fontana and Buss, 1993). In this constructivist approach organizations emerge from the collective behavior of primitive objects without any prior assumptions as to the nature of the objects or their kinetic coupling beyond that required by logical consistency with established physics and chemistry. We here elaborate the theoretical framework, modelling platform, and organizations emanating from this approach. The fact that organizations arise in a system lacking any formulation of Darwinian selection calls for increased attention to structural aspects of evolution.

2 The Structure of the Problem

Equations describing dynamical systems contain variables that represent entities of the physical world. The couplings among variables are given by a network of incidences representing the interactions between entities. A traditional dynamical system requires the explicit knowledge of this network of incidences. At the core of fundamental issues in evolutionary biology is the question: where does this network come from? To answer it, nonlinear dynamical systems must be extended in a fundamental way. *The underlying network of incidences (not just their strengths) must be subject to an endogenously induced dynamics, that is: the system must construct its own state space.* The present work provides a conceptualization and a solution to this problem.

We develop an approach where no particular fixed network is known *a priori*, by considering a finite ensemble in which interactions among objects repeatedly construct new objects. We will refer to such systems as *constructive dynamical systems*. In contrast to the traditional approach, a constructive dynamical system specifies the interactions among objects not externally, but rather internally to the objects as a function of their structure. A constructive dynamical system is, therefore, characterized by two components: a dynamics in phase space and a dynamics of the system's support. The term "support" refers to the changing dimensionality of the system. The notion is borrowed from mathematics, and denotes at any given moment the set of objects present in nonzero density at that time in the system.

We regard biological organizations as specialized systems of chemical transformation. Two specializations are foremost: the capacities for reproduction and for self-maintenance. Reproduction is what makes a system uniquely biological, in that it is this feature that triggers the Darwinian process. Self-maintenance is no less crucial, although its importance has been overshadowed by a traditional focus on reproduction and the associated Darwinian process. A self-maintaining chemical system is one which continuously regenerates itself by transformations internal to the system (Maturana and Varela, 1973, 1980). The emphasis is on the *relationships* amongst chemical species that permit continuous regeneration of the same *relationships*.

We have argued that two features of chemistry are essential to generate self-maintaining systems of transformation (Fontana and Buss, 1993). The first feature is the *constructive capability* of chemistry. This feature is reflected in the compositional syntax of molecules: molecules are either atoms or combinations of molecules. Construction taken alone, however, implies an ever-increasing diversity of molecules. In fact, molecules do not arbitrarily combine by juxtapostion. Their combination triggers specific structural rearrangements in accordance with the "laws of chemistry". Such rearrangements result in a set of specific stable forms. This second key feature induces a notion of *equivalence*, in the sense that multiple molecular combinations are effectively the same with regard to a particular product. In chemistry one can generate ethanol by reaction of acetaldehyde with sodium borohydride, or by reaction of an acetic acid ester with lithium aluminum hydride, or by a diversity of alternative routes. The syntactical structure <SODIUM BOROHYDRIDE APPLIED TO ACETALDEHYDE> is declared to be the same as the object <ETHANOL>. By this process the hydride/acetaldehyde combination is seen to be the same as the combinational object <LITHIUM ALUMINUM HYDRIDE APPLIED TO ACETIC ACID ESTER>. Similarly, pyruvate can be generated from glucose by glycolysis, or from the degradation of some amino acids. Construction permits diversity, and equality permits network formation.

These considerations inform the choice of formalism in which to embed a theory of the objects in question. The core idea of our abstraction is to consider a molecule to be a mathematical function and to consider its interaction upon collision with another molecule to be a functional application. This leads to an abstract chemistry. Not an arbitrary one, but rather one that is based on an analysis of the fundamental concept of "function". Such an analysis constitutes the foundation of computation. The need for a universe in which to express objects as functions, then, leads us directly to a natural representation of objects as elements in a functional language, more specifically, as words in λ-calculus (Church, 1932; Church, 1941; for monographs on λ-calculus with different perspectives see: Barendregt, 1981; Hindley and Seldin, 1986; Revesz, 1988; Huet, 1992). λ-objects share with molecules a recursive syntactical architecture. λ-objects can be syntactically juxtaposed, a process called *application*, to form new λ-objects. Such a combination triggers syntactical rearrangments that produce a unique normal form (if one exists), which we view as standing in a logical analogy to a "stable form." These rearrangements occur in accordance with a few fixed schemata, which constitute a procedure for evaluating the application of a function to an argument by formalizing the notion of substitution. These schemata are generic rewrite rules, and we view them as a logical counterpart to the "laws of chemistry".

Our focus is not on the behavior of any individual object, rather we seek to understand the many-body phenomena arising from a large number of such objects when they interact applicatively. In a finite system a problem arises when new objects can appear and disappear as a result of constructive interactions occurring within the system's support, particularly when the infinite universe of possible objects is the only *a priori* certain closure (Farmer *et al.* 1986, Varela *et al.*, 1988, Bagley *et al.*, 1989, 1992; Bagley and Farmer, 1992). We here simply frame the system as a stochastic simulation in which individual objects construct new objects directly on a collision by collision basis (Fontana and Buss, 1993).

The main distinguishing feature of a constructive dynamical system is the induction of an apparently deterministic "motion" (despite stochastic collisions and minor noise) of the system's support. Any particular such motion occurs because of the particular relationships of mutual construction at any given time. The addition (or the loss) of an object species may mean the addition (or the loss) of relations among objects in the system. Hence attention must be paid to both the dynamics of the set of objects as well as the dynamics of the functional relationships instantiated by it. An interesting possibility then arises: while the set of objects may continue to change, the relational structure may become invariant. The system has hit a "fixed point" in terms of relations among objects. An "attractor" of this kind is a well known mathematical object: in this case, the attractor is an algebraic structure.

We view a dynamically maintained algebraic structure as a minimal organization. Such an organization has kinetic, grammatical and functional boundaries which qualify it as an object capable of combining with other objects, thus giving rise to new organizational levels. We claim that a model addressing the emergence of such organizations offers a solution to the existence problem in evolutionary theory, and that the specific forms of these organizations invite a series of biological interpretations.

3 Guide to the Reader

Having introduced, in sections 1 and 2, the existence problem in evolutionary theory and why the structure of this problem necessitates the development of a constructive dynamical system, we briefly summarize the remaining chapters of this text as a guide to the prospective reader.

In section 4, we provide a primer to the λ-calculus to aid readers unfamiliar with this formalism, highlighting features of the calculus which are of particular importance for our approach. A reader for whom λ-calculus is novel will find this section and accompanying appendices essential. Having outlined the principle features of λ-calculus required to evaluate our claim, we return in the final subsection to the claim that it is a natural framework in which to represent the features we wish to abstract from chemistry. Readers familiar with λ-calculus will find this section of use.

In section 5, we introduce a variety of implementations developed to explore the collective behavior of λ-expressions. Three implementations are discussed, formost among which is the protocol for a stochastic flow reactor model upon which all experimental results are based. We also introduce a representation in terms of ordinary differential equations and an iterated set representation, in which the constructive and kinetic components of the flow reactor are disabled respectively. Both serve as an aid to the characterization of organizations that emerge in our experiments.

Section 6 presents our experimental results with the stochastic flow reactor. Our results fall into three natural groupings (labelled Levels 0, 1, and 2), which invite biological interpretations. We resist the temptation, however, to conflate biological interpretation with the presentation of results, and for each level describe in considerable detail the features of several organizations that emerge. The reader whose sole interest is in assessing the biological significance of our findings can likely be satisfied with a thorough understanding of a single example from each level and a careful reading of the summary of each section. The subsection 6.2.1, preceding the Level 1 findings, is of particular importance, as it is here that "what it means to be organized" is established concretely.

We conclude with two sections, the first of which (section 7) makes a modest beginning in placing limits on the important open theoretical question of defining the range of organizations that are possible and the limits to their mutability. In the discussion (section 8), we return to an assessment of the utility of the abstract chemistry we employ and explore the relationship of our work to the established research traditions (subsection 8.1). We then proceed to apply the findings of our experimental studies to the existence problem in evolutionary theory, with particular reference to the resemblance of the various levels of organization found in our experimental work to actual organizational transitions in the history of life and to the problem of the origin of variation (subsection 8.2). It is here that readers desiring a biological interpretation of the results of our experiments will find satisfaction. Finally, we conclude with a brief consideration of the extent to which the approach defined here may prove useful to other sciences in which organizational issues are prominent (subsection 8.3).

4 The Constructive Dynamical System. Part 1. The Calculus

4.1 A λ-calculus primer

λ-calculus[1] makes precise the syntactical process of evaluating a function F for a given argument x. The informal notation of $F(x)$ is ambiguous. As it stands, it is not clear if $F(x)$ denotes the *mapping* from x to $F(x)$, or if $F(x)$ denotes the *value* of the mapping F at x (that is: the application of F to x). Let us settle for

[1] Another simple conceptual introduction to λ-calculus may be found in Penrose (1989).

$F(x)$ meaning the value of F at x. To indicate the mapping we write $x \mapsto F(x)$. For example[2], $x \mapsto x \cdot x + y$.

λ-calculus is a clever notational scheme to keep this distinction unambiguous throughout symbolic manipulations. In λ-calculus we write for the application of the function F to x simply $(F)x$. This differs from usual notation only in the parentheses now delimiting the term that is being applied. The function in x, on the other hand, is designated by a "$\lambda x.$" prefix, which declares x to be a formal parameter of F: $\lambda x.F$. The application of the function $\lambda x.F$ to x reads as $(\lambda x.F)x$. For example: the λ-calculus version of the statement "apply the function $x \mapsto x \cdot x + y$ to 10" is "$(\lambda x.\, x \cdot x + y)10$." Designating F as a function in x *and* y is done by adding a corresponding λ prefix: $\lambda x.\lambda y.F$.

The designation "$\lambda x.$" defines x to be a variable, that is, a slot that can be occupied by *any* argument (including another function). The application of a function to an argument is carried out by *substitution*. When the substitution occurs, the designation "$\lambda x.$" is removed, because the slot has been occupied. For example: the application of the term $\lambda x.\, x{\cdot}x{+}y$ to 10, written as $(\lambda x.\, x{\cdot}x{+}y)10$, is executed by replacing every occurrence of x within the scope of the "$\lambda x.$"-marker by 10, while removing that marker together with the parentheses: $10 \cdot 10 + y$.

The informal notation $x \mapsto x \cdot x + y$ states that y is not considered as a placeholder to be substituted by an argument. y is intended to be arbitrary but otherwise fixed. Such a variable is called *free*. In contrast, a variable under the scope of a λ-designation is called *bound*. In λ-calculus *any* λ-term can replace a bound variable. The idea of substitution is to produce only tautologies. Stated informally: if, for example, $F : x \mapsto x^2$, then one may use "$F(2)$," or "$(x \mapsto x^2)$ applied to 2", or "2^2," or just "4" interchangeably; they mean the same thing.

Care is needed in preventing free variables from becoming bound during the substitution process, since this would not conserve meaning. We recognize, for example, that the actual name of x in $\int(x^2 + y)\,dx$ is not important; we may as well substitute z for x: $\int(x^2+y)\,dx = \int(z^2+y)\,dz$. However, substituting y for x yields $\int(y^2+y)\,dy$ which is not the same as $\int(x^2+y)\,dx$. Similarly in λ-caluclus: naive application of $A \equiv \lambda x.\lambda y.\, x \cdot x + y$ to $B \equiv y$ would produce $\lambda y.\, y \cdot y + y$. The variable y, which occured free in B, has been bound by the remaining "$\lambda y.$" of A. This is inconsistent, since the y in A and the y in B are really intended to be different names. To perform the application we must first rename the (bound) y in A to, say, $A \equiv \lambda x.\lambda z.\, x \cdot x + z$.

[2] In λ-calculus there are no *a priori* given primitives such as multiplication, addition, or even numbers; there are only the bare essentials: variables and substitution.

4.2 Axioms of λ-calculus

The following axiomatic definition is taken from Revesz (1988), and is similar to Curry's approach to substitution (Curry and Feys, 1958). Throughout this paper operations with λ-expressions follow *verbatim* this definition. The first three axioms define the syntax of terms. The remaining axioms formalize the notion of substitution by recursion over the structure of terms, avoiding any confusion of free and bound variables. Appendix A provides accompanying comments.

(i) Terms

λ-terms are words over an alphabet consisting of

- an infinite supply of variables: $V = \{x_1, x_2, \ldots\}$
- the abstraction symbol: λ
- structural symbols: "." (dot) and "(", ")" (parentheses)

The set of λ-terms, Λ, is defined inductively:

$$\begin{aligned} x \in V &\Rightarrow x \in \Lambda && (1)\\ x \in V, M \in \Lambda &\Rightarrow \lambda x.M \in \Lambda \quad \text{(abstraction)} && (2)\\ M \in \Lambda, N \in \Lambda &\Rightarrow (M)N \in \Lambda \quad \text{(application)} && (3) \end{aligned}$$

A variable x is said to be *bound* if it occurs inside a subterm with the form $\lambda x.P$, otherwise it is *free*. The set of free variables in an expression P is denoted by $\phi(P)$.

(ii) Syntactical Transformation

The schemes of transformation are oriented rewrite rules. Structures on the left-hand side are replaced by structures on the right-hand side. In the following upper case letters denote arbitrary terms, and lower case letters denote variables.

substitution

$$\begin{aligned} (\lambda x.x)Q &\longrightarrow Q && (4)\\ (\lambda x.E)Q &\longrightarrow E, \text{ if } x \notin \phi(E) && (5)\\ (\lambda x.\lambda y.E)Q &\longrightarrow \lambda y.(\lambda x.E)Q, \text{ if } x \neq y \text{ and } (x \notin \phi(E) \vee y \notin \phi(Q)) && (6)\\ (\lambda x.(E_1)E_2)Q &\longrightarrow ((\lambda x.E_1)Q)(\lambda x.E_2)Q && (7) \end{aligned}$$

renaming

$$\lambda x.E \longrightarrow \lambda z.(\lambda x.E)z, \; z \notin \phi(E) \qquad (8)$$

4.3 Terminology

To make the paper self-contained for readers unfamiliar with λ-calculus, we introduce some informal background. Details may be found in Barendregt (1981).

4.3.1 Normal form

A λ-term of the kind indicated on the left hand side of (4–7) is called a *redex*, and the corresponding right hand side is its *contractum*. A λ-term that contains no redex is called a *normal form*. The repeated operation of axioms (4–8) on a λ-term until it is in normal form, is called a *reduction*. A term need not have a normal form, as exemplified by $(\lambda x.(x)x)\lambda x.(x)x$ - the application of $\lambda x.(x)x$ to itself:

$$\begin{aligned}(\lambda x.(x)x)\lambda x.(x)x &\xrightarrow{(7)} ((\lambda x.x)\lambda x.(x)x)(\lambda x.x)\lambda x.(x)x \xrightarrow{(4)} \\ &(\lambda x.(x)x)(\lambda x.x)\lambda x.(x)x \xrightarrow{(4)} (\lambda x.(x)x)\lambda x.(x)x \xrightarrow{(7)} \text{ etc.}\end{aligned} \tag{9}$$

The process depicted in (9) is an infinite loop.

The Church-Rosser theorem (Church and Rosser, 1936) states that if a term has a normal form, that normal form is unique. The standardization theorem (Curry and Feys, 1958) states that if a λ-expression has a normal form, then there is a normal sequence of contractions which terminates. This sequence involves the contraction of the leftmost redex at each step. The procedures used here follow Revesz (1988). Appendix B gives a complete example of reduction.

There is no effective procedure for deciding if an arbitrary term has a normal form. This is the λ-calculus analogue to Turing's "Halting-Problem."

4.3.2 Maps

Each term is a mapping from Λ into Λ, which we may view in the following way: fix an arbitrary term $A \in \Lambda$ and apply it to a term $B \in \Lambda$. This yields - by axiom (3) - a new term $(A)B \in \Lambda$. Hence A is the map

$$A : \Lambda \to \Lambda,\ B \mapsto (A)B. \tag{10}$$

The term $(A)B$ may be reduced to normal form using the rearrangements (4–8). Reduction is like "carrying out the computation." Since computation in λ-calculus is just substitution, and since substitution conserves the applicative behavior, $(A)B$ has the same behavior as its normal form, say C. That is: $(A)B$ applied to D is the same as C applied to D, for all D. Hence $(A)B$ and C indicate the same behavior, which we identify with C.

4.3.3 Equivalence classes

Both reduction to normal form and renaming of bound variables induce equivalence relations on Λ. The fact that both preserve applicative behavior in the sense of (10), permits interpretation of these classes as sets of expressions that are equal in regard to the function they represent.

(i) Equivalence classes by renaming

Two terms that differ only in the names of their bound variables are in the same equivalence class. They can be transformed into one another by appropriate sequences of renaming operations (8). For example:

$$\begin{aligned} &\lambda x.\lambda y.((x)(y))x \xrightarrow{(8)} \lambda u.(\lambda x.\lambda y.((x)(y))x)u \to \overset{(\cdots)}{\cdots} \to \lambda u.\lambda y.((u)(y))u \\ &\xrightarrow{(8)} \lambda u.\lambda v.(\lambda y.((u)(y))u)v \to \overset{(\cdots)}{\cdots} \to \lambda u.\lambda v.((u)(v))u \end{aligned} \tag{11}$$

where the second use of (8) involved the subterm $\lambda y.((u)(y))u$.

(ii) Equivalence classes by normal form

Two terms that reduce to the same normal form (modulo renaming of bound variables) are members of the same equivalence class.

4.4 Model-specific assumptions

We use λ-calculus in a way that is specific to our model intentions.

1. All terms are in normal form

   All expressions in our model universe must be in normal form under the conditions stated in the next item.

2. Pragmatic reduction

   To reduce a λ-expression only limited amounts of time and space can be used. A time unit is considered as one reduction step, that is: usage of one axiom (4–7), excluding renaming. A space unit is a character. Expressions that do not acquire a normal form within preset limits (section 5.3) are barred from our model universe. Our universe, therefore, has no halting problem.

3. Identification of terms modulo renaming

   Terms that are equivalent with respect to renaming of bound variables are identified. The representative of an equivalence class is obtained by systematically renaming all bound variables in a term to standardized names consisting of an x indexed by the order in which its binding λ occurs. For example:

$$\lambda x.\lambda u.\lambda c.((u)(x)a)\lambda a.(c)a = \lambda x_1.\lambda x_2.\lambda x_3.((x_2)(x_1)a)\lambda x_4.(x_3)x_4. \quad (12)$$

4. Identification of terms modulo normal form

   Since all expressions are in standardized normal form, our model universe is the quotient set of Λ with regard to the equivalence relations induced by renaming and (pragmatic) reduction.

4.5 The choice of λ-calculus

As in every foundational attempt, a formalism functions as both an expression and justification of a level of description. Having outlined the basics of λ-calculus, we return to our claim that it stands in analogy to chemistry. Chemistry is the science that defines how substances can be transformed into other substances. Our stylized "chemistry" is a space of objects, Λ, such that each object in Λ is a map from Λ into itself.

Further, we note that when two molecules interact, the product is determined by their structure, i.e., the components of which they are built and the manner in which these components are arranged. Thus, a molecule is an object with both a syntactic structure and an implied function. Syntactically, it is built up from component objects, according to well-defined rules. Its function, coded by its structure, is revealed by the chemical reactions in which it partakes. The object/function duality of molecules is precisely the ambiguity that λ-calculus clarifies.

For *motivational* purposes we place the structure of λ-expressions in correspondence to molecular structure, the risk of suggesting an inappropriate relation between the two notwithstanding. An aliphatic carbon chain is like a λ-expression with no abstractions. The introduction of a hydroxy group, for example, enhances the reactivity of the molecule. The hydroxy group is "active" in the sense that it can be swapped for another molecule in a reaction. In fact, chemists talk literally of "substitution". Abstraction in λ-calculus performs a similar role: it declares a symbolic unit to be "swappable" (that is: substitutable) by another expression. The same holds for carbonyl groups, amino groups, double bonds and the like. Chemical language refers to them as "functional groups". Abstraction captures the role of functional groups embedded in the reactants.

The process of a chemical reaction starts by placing the reactants in proper juxtaposition. This forms a "transition state" which is just another molecule that quickly undergoes structural rearrangements eventually resulting in a stable product. In λ-calculus a new expression is formed by the juxtaposition of two expressions, a syntactic operation called application. The new expression, like the transition state, can be rearranged in accordance with the axioms of reduction, (4)–(7). Application, thus, captures the combination of molecules to generate a transient molecule. The chemical product resulting upon electronic rearrangements is determined by the way functional groups are embedded in the transition state. Similarly, the normal form of a λ-expression is determined by the relative positions of the abstractions occuring in it. Of course, we have to require the execution of the appropriate syntactical transformations until normal form is reached. In chemistry this happens spontaneously because of an energetic driving.

The reduction schemes of λ-calculus represent a theory of equality, allowing determination of when two applications of λ-expressions are the same. Rearrangement schemata, which state what can be transformed into what, are hardly similar in chemistry and computation. While reductions can be defined in various ways (Newman, 1941), they require an order relation in the space of objects. Thermodynamics can be viewed as a device which induces an order relation among molecular objects, thereby enabling a process of reduction (in a mathematical sense) to take place automatically.

While λ-calculus captures precisely those features we seek to abstract from chemistry - i.e., the construction of new molecules upon reaction and the equivalence of reactions with respect to products - it should not be confused with actual chemistry. A λ-calculus based "chemistry" is competent to explore those logical possibilities that follow from these two features of chemistry, but no others. Hence, we should not expect λ-expressions to *emulate* particular molecules - we should not, for example, expect a self-replication to be equally easy to obtain in actual chemistry as it is with λ-expressions. In standing as a surrogate for chemistry, i.e., an abstract chemistry, it can do no more than inform us as to what can be generated from the abstracted features.

There are many alternative approaches to computation which are formally equivalent to λ-calculus, raising the question of why we chose Λ. Each of these alternative approaches, however, represents formalizations of fundamentally different intuitive models (e.g., Turing machines rest on the notion of a state transition, λ-calculus rests on the notion of substitution, Post (or Thue) systems are aimed at recursive enumerability, distributed state transitions are central to cellular automata, and so forth; for an overview see Odifreddi, 1989). While it is reassuring that all these irreducible models do indeed express the same class of functions, they are far from being the same in any other regard. While all models of computation share construction, only λ-calculus is a natural theory of equivalence. These two features, constructivism and equivalence relations, are that which we hope to capture from chemistry. Independent of the chemical analogy we take the induction of equivalence relations through constructive processes to be the basic organizing concept behind any constructive dynamical system. While in the present work equivalence is in regard to function, other notions of equivalence can be captured by appropriate "calculi". However, for the extension of nonlinear dynamical systems within the scope of the chemical analogy λ-calculus is as much the proper formal structure as differential (or difference) equations are for the definition of dynamical systems.

5 The Constructive Dynamical System. Part 2. Implementations

We seek to explore the collective behavior of λ-expressions in a minimal kinetic model environment based on mass action (Fontana, 1990, 1991, 1992; Fontana and Buss, 1993). λ-expressions are consequently treated as if they were physical particles. We will henceforth refer to them as objects or object species that can occur in multiple instances. In this environment space, free energy, conservation laws, or any of a number of plausible specializations are absent. This minimal model can be regarded as a tool for studying the implications of the aforementioned abstractions from chemistry.

5.1 Definitions

To set the dynamical stage we will need two definitions. (i) What happens when two objects (i.e., λ-expressions) collide? We refer to this definition as the *collision rule*. (ii) What happens with these objects after a collision, more precisely: what is the overall balance equation? We refer to the corresponding definition as the *reaction scheme*.

The collision rule

Let $\circ$ denote the action between two objects A and B upon collision. The map $\circ : \ \Lambda \times \Lambda \to \Lambda$ sends an ordered pair of objects into a third object. In the simplest case the action is application, axiom (3), with subsequent reduction to normal form:

$$A \circ B : \Lambda \times \Lambda \to \Lambda, \ [A, B] \mapsto (A)B \overset{\text{reduction to normal form}}{\Rightarrow \cdots\cdots \Rightarrow \cdots\cdots \Rightarrow} C. \tag{13}$$

Collision between A and B creates a new function $(A)B$ which will be represented by its normal form object C,

While we utilize (13) as our standard collision rule, we occassionally employ a generalized collision rule. Note that $\circ$ in (13) can also be stated as the application of the λ-expression $\lambda x_1.\lambda x_2.(x_1)x_2$ to the arguments A and B. This suggests generalization to an arbitrary λ-expression that mediates the collision between A and B. We denote this expression by Φ, and the corresponding action as $A \circ_\Phi B$:

$$A \circ_\Phi B \equiv ((\Phi)A)B \overset{\text{reduction to normal form}}{\Rightarrow \cdots\cdots \Rightarrow \cdots\cdots \Rightarrow} C \tag{14}$$

When Φ expresses application as in (13), we omit the index from $\circ$.

The generalized collision law has Φ transforming A into some A' (a translation of Λ) prior to its action on B, thus effectively preprocessing it: A behaves as if it were A'. For example, the mediation of the collision between A and B by $\Phi \equiv \lambda x.\lambda y.((\Phi_1)x)(\Phi_2)y$, with $\Phi_1, \Phi_2 \in \Lambda$ yields $((\Phi_1)A)(\Phi_2)B$ as an intermediate step during reduction. Hence it can be regarded as a transformation of both A and B prior to their applicative action.

The reaction scheme

The reaction scheme defines the overall balance equation. The main assumption we make is that the colliding objects are not consumed in the collision process. At the microlevel this reads as:

$$A \circ_\Phi B \longrightarrow \text{nf of } ((\Phi)A)B \; + \; A \; + \; B, \tag{15}$$

where nf is shorthand for normal form.

The reaction scheme (15) is not essential. It can be relaxed to the case of a catalytic transformation:

$$A \circ_\Phi B \longrightarrow \text{nf of } ((\Phi)A)B \; + \; A. \tag{16}$$

under either of two conditions: (i) if the system is initialized with multiple instances of each object or if (ii) self-maintaining closure (see below) has occurred. We initialize our systems with a maximum diversity of objects, each present in a single copy, and, therefore, retain the reaction scheme (15) for system start-up (see section 6.2.6 for a discussion). We are aware that biological systems involve chemical reactions that are not catalytic (e.g., condensation reactions, etc.), but limit ourselves to the above reaction scheme as the appropriate first step.

Note that equation (15) defines a microevent which is asymmetric. The action of A on B is usually different from the action of B on A. However, when the collision partners are chosen at random, the two actions occur each with probability 0.5. Thus, written as a chemical reaction the scheme is symmetric:

$$A \; + \; B \begin{cases} \xrightarrow{0.5} \; A \circ_\Phi B \; \longrightarrow \; \text{nf of } ((\Phi)A)B \; + \; A \; + \; B \\[2ex] \xrightarrow{0.5} \; B \circ_\Phi A \; \longrightarrow \; \text{nf of } ((\Phi)B)A \; + \; A \; + \; B \end{cases} \tag{17}$$

If pragmatic reduction in (15) does not yield a normal form for $((\Phi)A)B$, then $((\Phi)A)B$ is null, and the collision is termed *elastic*. Otherwise the collision is *reactive*.

5.2 The flow reactor equations

We will consider a finite ensemble of objects that randomly collide according to (15). Reactive collisions make the system grow. We constrain the reactor to a constant number of particles by an unspecific dilution flow. Each time a new particle has been created by a reactive collision, one particle is chosen at random and is eliminated. Object species are, therefore, flushed out in proportion to the number of copies with which they are present in the system. The constraint on the total number of particles limits the lifetime (residence time) of each object. Despite the fact that reactively colliding objects are retained by (15), they cannot be maintained in the long run unless they are produced by some pathway.

It is instructive to consider the differential equations that model such a system, in the case of both real valued concentrations and closure with respect to interactions occurring within the system. The equations are most simply formulated in terms of the relative concentration of object species i, x_i, with $0 \leq x_i \leq 1$, and $\sum_k x_k = 1$.

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \sum_j \sum_k a^i_{j,k} x_j x_k - x_i \sum_{r,s,t} a^t_{r,s} x_r x_s, \quad i = 1, 2, \ldots \tag{18}$$

with

$$a^i_{j,k} = \begin{cases} 1 & \text{if } j \circ_\Phi k \Rightarrow i \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $\Rightarrow$ denotes pragmatic reduction.

A collision consists in applying a function, and yields a unique outcome. Hence there is at most one i for any given pair (j, k) such that $a^i_{j,k} = 1$. If all collisions are reactive, then $\sum_t a^t_{r,s} = 1$, and equation (18) simplifies to:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \sum_j \sum_k a^i_{j,k} x_j x_k - x_i, \quad i = 1, 2, \ldots \tag{20}$$

with $a^i_{j,k}$ as in (19). The first term collects all production pathways of i. The second term represents the proportional dilution flux, which keeps the system normalized at $\sum_k x_k = 1$. Equation (18) is appropriate for the generic case, when the rate coefficients $a^i_{j,k}$ are non-negative real numbers, when some collisions are elastic, or when collisions generate more than one product. Equation (17) is reflected in (18) by its invariance to a symmetric or asymmetric choice of the $a^i_{j,k}$. The typical behavior of (18) for random choices of the $a^i_{j,k}$ is very robust: one single globally stable fixed point in the interior of the concentration simplex (Stadler *et al.*, 1993).

Note that equation (18) has an important special case. Consider the case where the only allowed action between any two objects is a copy-action (i.e., reproduction), that is: $j \circ_\Phi k \Rightarrow j$ or k. Then $a^i_{j,k} = 0$ iff $i \neq j \vee i \neq k$, and using the notation $b_{i,j} = a^i_{i,j} + a^i_{j,i}$, equation (18) becomes

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = x_i \left(\sum_j b_{i,j} x_j - \sum_{r,s} b_{r,s} x_r x_s \right), \quad i = 1, 2, \ldots \tag{21}$$

which is the well-known replicator equation. The replicator equation appears in a variety of applications (for a review see Hofbauer and Sigmund, 1988) ranging from evolutionary game dynamics (Maynard-Smith, 1982; Lindgren, 1992) to models of molecular evolution (Eigen, 1971, Eigen and Schuster, 1979), and to ecology (Lotka, 1925; May, 1976). The neo-Darwinian framework is thus seen as a limiting case embedded in the broader context of transformation systems. It is this larger context that we hold as enabling us to address the genesis of organizational grades.

5.3 Protocol

Our system is a stochastic process of which equation (18) is an (imperfect) model. The standard experimental protocol follows.

1. Choose a collision rule Φ.

2. Initialize

   Generate N random expressions. A random expression is produced by recursively generating random subexpressions. With probability p_1 an atomic expression - axiom (1) - is generated; it consists of a variable from some finite set. With probability p_2 an abstraction - axiom (2) - is produced, and with probability p_3 an application - axiom (3). When a maximum nesting level (here: 20) has been reached, an atomic expression is chosen to avoid further recursion. Obviously, $p_1 + p_2 + p_3 = 1$. In some cases the probabilities p_i are also a function of the nesting level.

   Every expression so generated is reduced to standardized normal form. All initial expressions are unique in the population. $N = 1000$, if not otherwise stated.

3. Choose a random pair

   Choose two of the N objects at random. Let the first chosen object be j, and the second one k.

4. Reduce

   Compute the standardized normal form, i, of $j \circ_\Phi k \equiv ((\Phi)j)k$, according to pragmatic reduction.

   Computational limits involve space (total size in characters during any reduction step) and time (total number of invoked axioms, excluding renaming). If not otherwise stated the space limit is 4000 characters, and the time limit is 10,000 reduction steps. Despite an upper bound to the length of any expression implied by these limits, we explicitly limit the size of a standarized normal form to 1000 characters.

   If no normal form is produced within these limits, the collision is elastic, and the process continues with step 3.

5. Apply boundary conditions

   Filter the product object i by applying, if any, functional and/or syntactical boundary conditions.

   Functional boundary conditions impose constraints on the execution of particular actions, such as identity (copy) actions, or any other pattern of syntactical manipulations.

   Syntactical boundary conditions (*filters*) impose constraints on how objects are structured in terms of syntactical patterns. In particular, one boundary condition is *standard* for all our experiments. We require that all objects in operator position be abstraction forms, i.e., begin with a λ. The reason for this restriction is that the application of a normal form which does not begin with a λ generates an expression that cannot be reduced, hence belongs to a degenerate equivalence class which contains it as the only member. For example, $(A)B$ applied to C yields the normal form $((A)B)C$ which can be produced only by this collision. The applicative action of $((A)B)C$ generates further instances of this kind. Allowing this to occur would eventually eliminate any possibility for generating *transformation networks*.

   If the product i does not comply with the boundary conditions, the collision is elastic, and the process continues with step 3.

6. Apply flow and add product

   If the collision is reactive, one object, chosen randomly among the N present in the system, is eliminated. The collision product i is added to the system.

7. Iterate

   The process continues with step 3.

5.4 Iterated set actions

We may switch off the kinetic (mass action) aspect of our model by considering only iterated actions within a set of object species. Doing so permits us to introduce concepts that will prove useful in later discussion.

We proceed to define a few simple set valued iterated maps. Let $\mathcal{J}$ and $\mathcal{K}$ be subsets of Λ. The action of $\mathcal{J}$ on $\mathcal{K}$ is the set $\mathcal{I} \subseteq \Lambda$ defined as

$$\mathcal{I} = \mathcal{J} \circ_\Phi \mathcal{K} \equiv \{i \mid ((\Phi)j)k \Rightarrow i,\ (j,k) \in \mathcal{J} \times \mathcal{K}\}, \tag{22}$$

where we take $\circ_\Phi$ to include the application of functional and syntactical boundary conditions (see section 5.3).

Two set valued iterated maps are of special interest. The first is a "quadratic" map that replaces the old set at each iteration. Let $\mathcal{A} \subset \Lambda$ be an initially given set.

$$\begin{aligned} \mathcal{A}_0 &= \mathcal{A} \\ \mathcal{A}_{i+1} &= \mathbf{m}(\mathcal{A}_i) := \mathcal{A}_i \circ_\Phi \mathcal{A}_i, \quad i = 0,1,2,\ldots \end{aligned} \tag{23}$$

We will also write

$$\mathcal{A}_{i+1} = \mathbf{m}(\mathcal{A}_i) = \mathbf{m}^{i+1}(\mathcal{A}) = \underbrace{\mathbf{m}(\mathbf{m}(\ldots\mathbf{m}}_{i+1}(\mathcal{A})\ldots)). \tag{24}$$

A variant of (23) keeps the previous set. This turns out to be equivalent to refreshing the initial set $\mathcal{A}$ at each iteration:

$$\begin{aligned} \mathcal{A}_0 &= \mathcal{A} \\ \mathcal{A}_{i+1} &= \mathbf{M}(\mathcal{A}_i) := (\mathcal{A}_i \circ_\Phi \mathcal{A}_i) \cup \mathcal{A}_i = \\ &= (\mathcal{A}_i \circ_\Phi \mathcal{A}_i) \cup \mathcal{A} = \mathbf{m}(\mathcal{A}_i) \cup \mathcal{A}, \quad i = 0,1,2,\ldots \end{aligned} \tag{25}$$

The replacement map $\mathbf{m}(\mathcal{A}_i)$ generates only particular terms among those that result from $2^{i+1} - 1$ collision events among members in $\mathcal{A}$. In contrast, consider the following sequence of sets:

$$\begin{aligned} s_0(\mathcal{A}) &= \mathcal{A} \\ s_n(\mathcal{A}) &= \bigcup_{0 \le i \le n-1} s_i(\mathcal{A}) \circ_\Phi s_{n-i-1}(\mathcal{A}), \quad n = 1,2,3,\ldots. \end{aligned} \tag{26}$$

The n-th set in the sequence contains all terms that can be generated by exactly n collisions among elements of $\mathcal{A}$, and

$$S_n(\mathcal{A}) \quad = \bigcup_{0\leq i\leq n} s_i(\mathcal{A}), \quad n = 0, 1, 2, \ldots \tag{27}$$

is the sequence of sets of all terms produced by up to n collisions in $\mathcal{A}$. The increasing sequence (27) generates the object horizon that is in principle accessible from the initial set of objects. In fact, the closure $\mathcal{A}^*$ of $\mathcal{A}$ (with respect to the collision rule Φ) is defined as:

$$\lim_{n\to\infty} S_n(\mathcal{A}) = \bigcup_{i=1}^{\infty} S_i(\mathcal{A}) = \mathcal{A}^* \subseteq \Lambda, \tag{28}$$

where Λ is countably infinite by construction (section 4.2).

Notice that

$$\forall\, n,\ S_n(\mathcal{A}) \subseteq \mathbf{M}^n(\mathcal{A}) \subset S_{2^n}(\mathcal{A}). \tag{29}$$

Since S_n, $\mathbf{M}^n$, and S_{2^n} are all increasing, and since

$$\lim_{n\to\infty} S_{2^n} = \bigcup_{n=1}^{\infty} S_{2^n} = \bigcup_{n=1}^{\infty} \bigcup_{k=1}^{2^n} S_k = \bigcup_{l=1}^{\infty} S_l = \mathcal{A}^*, \tag{30}$$

it follows that

$$\lim_{n\to\infty} \mathbf{M}^n(\mathcal{A}) = \mathcal{A}^*. \tag{31}$$

Iteration of the process (25) yields the closure of the initial set.

For later use we introduce the following definitions. Any set $\mathcal{B} \subset \mathcal{A}^*$, for which $\lim_{n\to\infty} \mathbf{M}^n(\mathcal{B}) = \mathcal{A}^*$, is termed a *generator* of $\mathcal{A}^*$. Any set $\mathcal{B} \subset \mathcal{A}^*$, for which $\lim_{n\to\infty} \mathbf{m}^n(\mathcal{B}) = \mathcal{A}^*$, is termed a *seeding set* of $\mathcal{A}^*$. These definitions emphasize a distinction. There are subsets of the closure for which the replacement map $\mathbf{m}$ behaves effectively like the cumulative map $\mathbf{M}$. For a seeding set $\mathcal{B}$ to generate the closure $\mathcal{A}^*$ under the replacement map $\mathbf{m}$, that set must clearly be regenerated under $\mathbf{m}$ at some point.

$\mathbf{m}$ exhibits an interesting behavior under circumstances which are frequently realized in the dynamically generated closures discussed in the next chapter. Suppose that $\mathcal{B}$ is a finite seeding set of an infinite $\mathcal{A}^*$, and that there exists a finite l' such that $\mathcal{B} \subset \mathbf{m}^{l'}(\mathcal{B})$. Let l be the smallest such l'. It then follows that $\mathbf{m}(\mathcal{B}) \subset \mathbf{m}^{l+1}(\mathcal{B})$. Hence,

$$\begin{array}{ccccccccc}
\mathbf{m}^0(\mathcal{B}) & \subset & \mathbf{m}^l(\mathcal{B}) & \subset & \mathbf{m}^{2l}(\mathcal{B}) & \subset & \mathbf{m}^{3l}(\mathcal{B}) & \subset & \cdots \\
\mathbf{m}^1(\mathcal{B}) & \subset & \mathbf{m}^{l+1}(\mathcal{B}) & \subset & \mathbf{m}^{2l+1}(\mathcal{B}) & \subset & \mathbf{m}^{3l+1}(\mathcal{B}) & \subset & \cdots \\
\mathbf{m}^2(\mathcal{B}) & \subset & \mathbf{m}^{l+2}(\mathcal{B}) & \subset & \mathbf{m}^{2l+2}(\mathcal{B}) & \subset & \mathbf{m}^{3l+2}(\mathcal{B}) & \subset & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{m}^{l-1}(\mathcal{B}) & \subset & \mathbf{m}^{2l-1}(\mathcal{B}) & \subset & \mathbf{m}^{3l-1}(\mathcal{B}) & \subset & \mathbf{m}^{4l-1}(\mathcal{B}) & \subset & \cdots
\end{array} \tag{32}$$

In the case of a $\mathcal{B}$ for which $l = 1$, the trajectories of $\mathbf{m}(\mathcal{B})$ and $\mathbf{M}(\mathcal{B})$ are identical and monotonically non-decreasing. For a seeding set $\mathcal{B}$ with $l > 1$ the stroboscopic view of the trajectory $\mathbf{m}^t(\mathcal{B}), t = 0, 1, \ldots$ at $t = kl + m, k = 0, 1, \ldots$ shows for any particular $m \geq 0$ a monotonically non-decreasing subseries, equation (32). Moreover, for the unions $\mathbf{u}_k = \bigcup_{kl \leq i < (k+1)l} \mathbf{m}^i(\mathcal{B})$ for $k = 0, 1, \ldots$ of the trajectory segments from $i = kl$ to $i = (k+1)l - 1$ we also have $\forall\ k,\ \mathbf{u}_k \subset \mathbf{u}_{k+1}$. Hence, the trajectory "engulfs" itself in such a way as to always contain at time t all sets produced at epochs $t - kl$ for $k = 1, 2, \ldots \lfloor t/l \rfloor$. Since the entire series converges to $\mathcal{A}^*$, and since each set on the trajectory is contained in $\mathcal{A}^*$ by definition (28), one expects more and more overlaps within a "period" eventually to occur. When some set $\mathbf{m}^{kl+m_1}(\mathcal{B})$ is contained in a set $\mathbf{m}^{kl+m_2}(\mathcal{B})$ with $m_1 < m_2 < l$ the "period" l will shorten. For a finite seeding set $\mathcal{B}$ and infinite $\mathcal{A}^*$ there may eventually be a finite $k > 1$ such that $\mathbf{m}^i(\mathcal{B}) \subset \mathbf{m}^{i+1}(\mathcal{B})$ for $i \geq k$.

Finally we note that there can be initial sets which yield the closure $\mathcal{A}^*$ under $\mathbf{m}$, and which are not subsets of $\mathcal{A}^*$. The totality of these sets constitutes a kind of "basin of attraction" for $\mathcal{A}^*$. We want to distinguish the collection of such sets from the collection of seeding sets, because the latter are associated with a kind of "stability" of $\mathcal{A}^*$ under $\mathbf{m}$ with respect to the removal of random subsets. Clearly, a rigorous discussion of the range of possible "dynamical" behaviors of $\mathbf{m}$ under arbitrary initial conditions is desirable, but requires a suitable topology which is not yet available.

Under $\mathbf{m}$ the object species of iteration $l+1$ are not guaranteed ever to meet all object species of iteration l, as is the case with $\mathbf{M}$. The iterated map framework proves to be conceptually useful, despite the fact that our model system is an asynchronous stochastic process (section 5.3) where the iterations undergone by an object species are frequency dependent.

6 Computer-generated organizations

We describe a variety of computer experiments carried out according to the protocol described in section 5.3. These experiments yield results that are naturally categorizable into one of three classes: (i) ensembles of reproducing objects, (ii) self-maintaining organizations, and (iii) metaorganizations composed of self-maintaining organizations.

6.1 Level 0 - reproduction and ecology

A typical flow-reactor experiment starts with an initial set of random objects, without any further boundary conditions, and with application as the collision rule. The typical outcome is a small set of objects, $\mathcal{O}$, which is (i) closed with respect to interaction, and where (ii) every object is engaged in a copy-action. During an early phase, collisions among objects frequently produce objects that are new to the system. Property (i) means that this innovatory activity ceases, and objects only produce objects that are already present. Property (ii) means that

$$\forall\, f \in \mathcal{O}\ \exists\, g \in \mathcal{O},\ \ g \circ f = f \text{ or } f \circ g = f. \tag{33}$$

Stated differently, each object is either a left or right fixed point of some other object, possibly including itself, in which case the object is a self-copier, $f \circ f = f$. In many instances the system reduces to just one object species that is a self-copier. In other cases the system contains a small stable ecology of objects engaged in mutual copy-actions. Figure 1 shows two typical examples.

We refer to this phenomenology as *Level 0* or L0. We will also refer to λ-expressions as L0 objects, because they constitute the relevant individuals in L0. The L0-phenomenology is fundamentally shaped by copy interactions. The structure of L0-ensembles recalls a stylized ecology consisting of cooperating individual replicators. These are λ-expressions, and as such they are located at the most primitive object level in our model.

The most trivial example of a copier (and self-copier) is the identity $\lambda x.x$. There are, of course, many less trivial ones. An important lesson comes from the fact that an object may be an identity only on a specific subset of the universe Λ, and behave differently outside that subset. Consider, for example, object B in the two-membered ecology of Figure 1(a). Both B and A are fixed points of B. But if B were to encounter object C of Figure 1(b), then $B \circ C = \lambda x_1.\lambda x_2.(x_2)\lambda x_3.\lambda x_4.\lambda x_5.(x_5)x_4 \neq C$. The structure-function relations that define our objects imply the possibility of specific roles, particularly kinetic ones, within the specific context of other objects.

Most L0-ensembles have the form of n-membered elementary hypercycles (Eigen, 1971; Eigen and Schuster, 1977, 1978a, 1978b, 1979), where the major cooperative backbone consists of object $i \bmod n$ copying object $(i+1) \bmod n$. L0-ensembles are typically not robust towards functional perturbations. When a small number of random objects is introduced into the system, L0-ensembles typically collapse to a single replicator. A framework for the study of L0 is provided by equations of the replicator, (21), or Lotka-Volterra type (Hofbauer and Sigmund, 1988). The present framework, thus, reproduces well known theoretical ground.

6.2 Level 1 - self-maintenance and organization

When we eliminate copy-actions, our results differ markedly. To prevent copy reactions we apply a functional boundary condition on the flow reactor, such

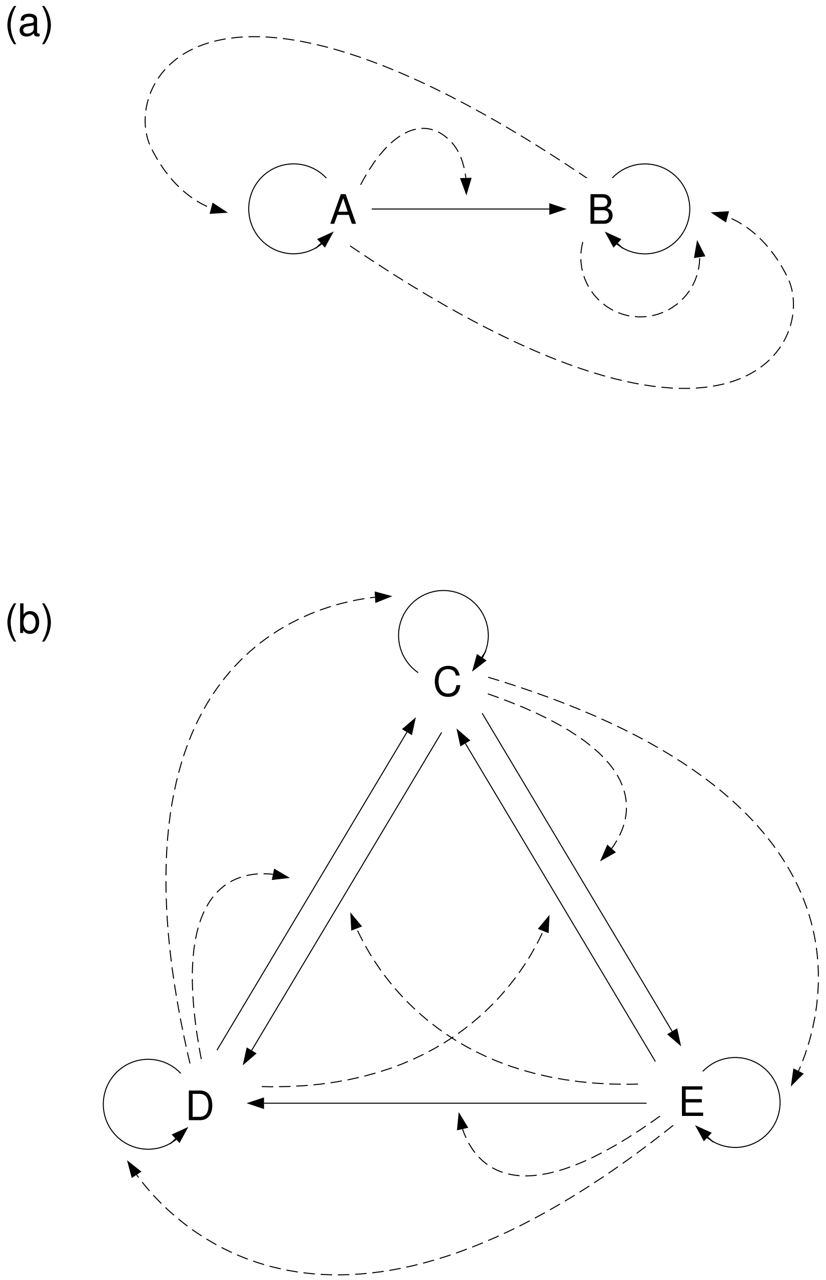

Figure 1: Level 0 ecology. A solid arrow denotes a transformation from argument object to value object. A dotted arrow connects the operator object with the transformation performed by it. Thus, $E \circ C = D$ is shown as a solid arrow from C to D pointed at by a dotted arrow originating in E. $A \equiv \lambda x_1.(x_1)\lambda x_2.\lambda x_3.(x_3)\lambda x_4.\lambda x_5.(x_5)x_4$, $B \equiv \lambda x_1.(x_1)\lambda x_2.\lambda x_3.(x_3)x_2$, $C \equiv \lambda x_1.\lambda x_2.(x_1)\lambda x_3.x_1$, $D \equiv \lambda x_1.\lambda x_2.\lambda x_3.(x_2)\lambda x_4.x_2$, $E \equiv \lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.(x_3)\lambda x_5.x_3$

that if a collision results in a product that is syntactically identical to any of the objects involved in the collision, that collision is declared as "elastic", and the product is ignored (e.g., we eliminate from our system zero-length cycles that characterize the interaction graphs of Figure 1).

The complete elimination of copying generates self-maintaining organizations. To grasp what we mean by an organization we first introduce an informal discussion of how an observer might discover that the behavior displayed in the flow reactor has become "organized." We follow this discussion with a series of specific examples which we will subsequently utilize to identify the properties defining an organization.

6.2.1 Organizations, algebraic structures and observers

Let us now adopt the viewpoint of someone observing our stochastic flow reactor. We shall assume that the observer knows nothing of the λ-calculus, and hence is ignorant about the microscopic mechanics of our model universe. The observer will essentially see random collisions between seemingly meaningless symbolic structures. The content of the reactor changes over time as new objects are produced and others disappear.

The observer may try to understand the system by detecting regularities. This can be done by (i) analyzing the structures to identify common syntactical patterns, and by (ii) analyzing all pairwise interactions to identify regularities among the actions between structures. In other words, the observer will try to describe the system in terms of syntactical and of functional patterns. If there are no such regularities the system will hardly be perceived as "organized." The system's description will amount to a list of all structures, together with a list of all pairwise actions, stating who acts on whom to produce whom.

Suppose that our observer is staring at a reactor which we have prepared to contain symbolic structures that are a representation of the integers, such that their reactive collisions effectively amount to an addition operation under the current collision rule. With some naiveté the observer might soon detect regularities in the syntax of the individual structures (a grammar), as well as regularities and symmetries in the actions of the structures upon one another. The observer might discover, for example, that $\forall\ a, b, c \ \ (a \circ_\Phi b) \circ_\Phi c = a \circ_\Phi (b \circ_\Phi c)$. In particular, the observer might discover that the syntactical and the functional regularities characterizing the individual structures and their actions are invariant, that is, while interactions may yield new objects, any new objects will conform in syntax and behavior with the discovered regularities.

The laws of an algebraic structure may in part refer to the internal structure of the elements, and thus relate *subsets* of elements to one another. This leaves room for an entire spectrum of compressions up to the point where each object is specified completely, and hence the algebraic structure degenerates into an exhaustive tabulation of the mapping from $\mathcal{A} \times \mathcal{A}$ into $\mathcal{A}$. *The observer will conclude that the system is an organization to the extent that there is a compressed description of all of its objects and all of their relations.*

With some ingenuity the observer will further derive all laws implied by the uncovered group structure (Knuth and Bendix, 1970; Huet and Oppen, 1980). If read as rewrite rules, the equations thus obtained will enable the observer to exactly describe (and predict) each and every collision product in the system - without any knowledge about λ-calculus. The observer will, then, have discovered a perfectly valid theory of that organization, without reference to its underlying micromechanics.

Knowing, as we do, that the flow reactor is a many-body implementation of λ-calculus, we can deduce the same theory by using axioms (4–7) and induction over the particular grammatical structure of the objects in this organization. Restricted to this organization, the empirically discovered laws are equivalent to the λ-theory. The laws derived by the observer, however, constitute an *independent* level of description, with different symmetries from those existing at the level of full λ-calculus. It is in this sense, and only in this sense, that such laws can be taken to be *emergent* (Anderson, 1972; Baas, 1993). Organizations will be defined in terms of their emergent laws.

6.2.2 Example 1

The first example is the simplest organization that we have observed. All concepts introduced by way this example are generic to *all* organizations that we have observed in our reactor.

In this experiment, the collision rule was application, initial objects were random, copy-actions were declared elastic, and no further syntactical filters were imposed beyond the standard filter (section 5.3). The most immediate observation, which distinguishes the current situation from L0 experiments, is that the reactor does not close with respect to interaction. New objects are produced frequently. Eventually, however, one notices that the "new" objects have been in the system at an earlier time, but had been eliminated because of fluctuations in their numbers. The support of the system has effectively ceased to move around in the space of λ-objects. Despite settling in a particular region of Λ, it does not close. Moreover, increasing the capacity of the reactor does not result in closure. Alternating between the standpoint of our observer in section 6.2.1 and a limited knowledge of λ-calculus, we provide a syntactical and functional analysis of the reactor's behavior.

Syntactical characterization

At the syntactical level all objects in the system are recognized as conforming with a simple description. They can be identified with two indices:

$$A_{i,j} \equiv \lambda x_1.\lambda x_2.\cdots\lambda x_i.x_j, \quad j \leq i. \tag{34}$$

For example, $A_{3,1} \equiv \lambda x_1.\lambda x_2.\lambda x_3.x_1$. Let $\mathcal{O}_1$ denote the (infinite) set of all objects whose structure is patterned after (34). An object of $\mathcal{O}_1$ acts as a projection function, if it were supplied with the full number of arguments: applying $A_{i,j}$ to i arguments a_k, $k = 1, \ldots, i$ returns a_j: $(\ldots(A_{i,j})a_1)a_2)\ldots a_j)\ldots a_i \Rightarrow a_j$. However, the projection does not occur, since the collisions are binary. For example, the collision of projector $\lambda x_1.\lambda x_2.\lambda x_3.x_3$ with any other object $Q \in \mathcal{O}_1$ invokes axiom (5). Q is substituted for x_1, which does not appear in the body of the function. The λx_1.-designation is removed, and leaves after standardization the object: $\lambda x_1.\lambda x_2.x_2$.

Functional characterization

An analysis of all pairwise interactions among objects in the flow reactor reveals that the system can be completely described by two simple laws:

$$\begin{aligned} \forall\, i,j > 1, k, l \quad A_{i,j} \circ A_{k,l} &= A_{i-1,j-1} & (35)\\ \forall\, i \neq 1, k, l \quad A_{i,1} \circ A_{k,l} &= A_{k+i-1,l+i-1}. & (36) \end{aligned}$$

(The exclusion of a statement about the identity function $A_{1,1}$ comes from the artifical ban on copy actions.) The laws evidently imply closure of $\mathcal{O}_1$:

$$\forall P, Q \in \mathcal{O}_1 \quad P \circ Q \in \mathcal{O}_1 \tag{37}$$

In the present case the patterns of relationships (35, 36) can be rendered schematically in a plane (Figure 2), where every object is represented as a point at coordinates (j, i). Law (35) states that the operation of any object with $j \neq 1$ on any object yields the object represented by the point immediately below on the operator's diagonal. This law generates a flow of objects along each diagonal towards the $j = 1$ endpoint. Law (36) states that the object represented by the endpoint of the ith diagonal operates upon any argument such as to produce an object located at $i - 1$ points up the argument's diagonal. This reverses the flow of objects.

Independent description

The system departed from an initial set of random objects, and entered an invariant subspace $\mathcal{O}_1 \subset \Lambda$. This subspace is characterized by a particular syntactic description of its elements, and by a particular set of laws that describe all actions among them.

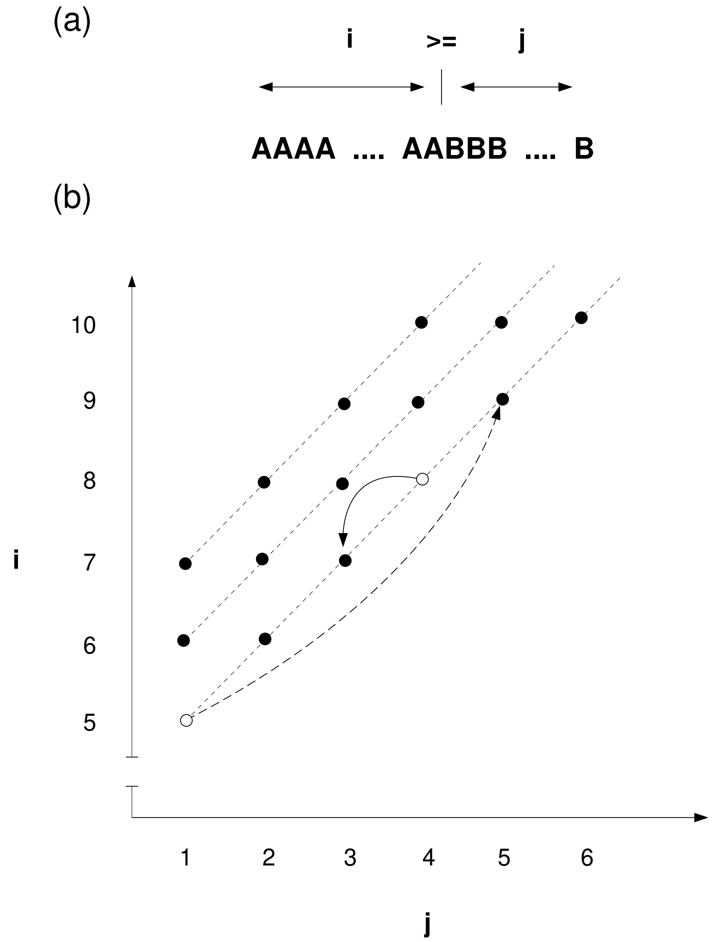

Figure 2: The simplest Level 1 organization. Objects retained in the system share a grammatical structure, figure 1(a). They can be represented as strings over two primitive building blocks **A** and **B**. The grammar requires a string to be a juxtaposition of two parts, the first containing only **A**'s, the second only **B**'s, such that there are at least as many **A**'s as **B**'s. In λ-notation **A** stands for an arbitrary abstraction, $\lambda x_i.$, and a sequence of j **B**'s represents the variable x_j. This illustrates a description of the system by means of an isomorphism that dispenses of λ-notation. As a result of its grammatical structure, every object can be represented as a point in the plane with coordinates (j, i), figure 1(b). Two laws (represented schematically) completely describe the relationships realized in this system. The first law states that the action of every object which is not the bottom object of a diagonal generates its lower neighbor on the diagonal independently of the argument object (solid line). The second law states that the bottom object on a diagonal generates particular objects located up the diagonal depending in a simple way on the argument object (dotted line). See text for details.

Note that both the syntactical and the functional descriptions of the organization are formulated independently from the underlying λ-calculus. The usage of λ-notation in (34) is for convenience, and does not require understanding of the symbolism. The point is that any object can be fully characterized by two indices, or by a grammatical structure, which may be represented in a variety of ways. Figure 2 is an example of a complete representation devoid of λ-notation.

Most importantly, the descriptions are complete in the sense that they contain everything that can be known about the system. The laws (35, 36) are a definition of reduction (Knuth and Bendix, 1970) for this particular subspace as characterized by the syntactical structure (34).

Self-maintenance

A set $\mathcal{A}$ of object species is *self-maintaining* if every object is produced by at least one interaction within $\mathcal{A}$; or in terms of set action (22), when

$$\mathcal{A} \subseteq \mathcal{A} \circ \mathcal{A}. \tag{38}$$

This implies that a self-maintaining subset is a seeding set (section 5.4) of the organization such that the iterates of the replacement map $\mathbf{m}$, (23), are monotonically non-decreasing, $\mathbf{m}_i(\mathcal{A}) \subseteq \mathbf{m}_{i+1}(\mathcal{A})$.

Center

The combined laws (35, 36) evidently ensure that the system is self-maintaining. Definition (38) suggests one more definition: the *center*, $\mathcal{C}$, of an organization $\mathcal{O}$ is its smallest self-maintaining subset:

$$\mathcal{C} = \min_{\mathcal{A} \subset \mathcal{O}} \{\mathcal{A} \subseteq \mathbf{m}(\mathcal{A})\}. \tag{39}$$

In the present case, to achieve self-maintenance the center must contain the $A_{i,1}$. The center is the smallest initial set for which the trajectories of the map $\mathbf{m}$ are identical to those of $\mathbf{M}$. While there usually are an infinity of smaller seeding sets, all organizations that we have observed possess a single unique center.

The kinetically maintained portion

A finite stochastic flow reactor will realize only a tiny subset of the invariant subspace. Consider for example a finite population of objects located far up one diagonal in Figure 2. Such a population is not self-maintaining. By virtue of law (35) it will slide downwards on the diagonal until it hits the bottom object that gives rise to law (36). Only at this point will objects upward on the diagonal be regenerated. As soon as the sample population has reached the center set, which includes objects $A_{i,1}$, it becomes self-maintaining. It is our observation that finite subsets of an organization move under flow reactor conditions towards the center, and subsequently remain anchored there. In the present case this is readily understood, since the overwhelming majority of reaction pathways leads down the diagonals. When the center is reached, the frequency distribution of objects in the reactor will stabilize. Of course, the portion of the organization that can be maintained under flow reactor conditions depends on the capacity of the reactor. There will, however, always be a sparsely populated periphery of object species that occasionally disappear by dilution, and occasionally are reconstructed by interactions within the remaining population.

Self-repair

Consider a fluctuation that removes a random subset $\mathcal{S}$ from the kinetically maintained portion of $\mathcal{O}_1$. The remainder will recover from the loss and repair itself back to the original state if the damaged portion contains a seeding set. Hence, the more seeding sets an organization can maintain under flow reactor conditions, and the smaller they are, the more robust it will be against random damage. Except for the trivial case of an object which is its own fixed point (an identity, for example) there can be no organization where every element is a seeding set, but it may be the case that every element is a generator.

Flow reactor conditions impose weaker requirements on self-repair than the replacement map $\mathbf{m}$ of section 5.4. Some generator sets, and not solely seeding sets, will determine the self-repair capabilities of an organization.

Kinetic persistence

Fluctuations in the composition of the support are caused by stochastic events where new objects are created in collisions and others are removed by the outflow. As a stochastic process our system has absorbing barriers which are of two types: (i) an inactive state composed of object species that collide only elastically, (ii) an active state in which every object species is a generator. L1-organizations show a remarkable degree of kinetic persistence, even when they are not themselves an absorbing barrier of type (ii). Some L1-organizations may, thus, be seen as extremely long lived metastable states which can "die," either because their intrinsic kinetics slowly drives them to a barrier of type (i), or because of some conflation of fluctuations from which the organization cannot recover (see previous paragraph on self-repair). Incidentally, in the present case each diagonal is an absorbing barrier of type (ii).

Self maintenance, (38), is a statement about the constructive capabilities of a set of objects, and is thus only a necessary but not sufficient condition for the kinetic persistence (Hofbauer and Sigmund, 1988) of an organization. The notion of organization refers to the grammatical and functional description of the system. It is this description which persists even when the set of objects continues to change (an example is described in section 6.3).

Families

The set $\mathcal{O}_1$ has a further fine-structure, which is visible in the reactor as well. $\mathcal{O}_1$ contains "families" defined as follows. Take an object $X \in \mathcal{O}_1$, and iterate the map $\mathbf{M}$ with X as the starting set. This process will generate a subset of $\mathcal{O}_1$. The subsets generated in this way by all objects in $\mathcal{O}_1$ are exactly the diagonals, indexed by i, in Figure 2:

$$\mathcal{A}_i = \{A_{i+l,1+l}, \ \ l = 0, 1, 2, \ldots\} \ \ i = 2, 3, \ldots \tag{40}$$

The $\mathcal{A}_i$ form a partition of $\mathcal{O}_1$: $\forall \ i \neq j \ \ \mathcal{A}_i \cap \mathcal{A}_j = \emptyset$, and $\mathcal{O}_1 = \bigcup_i \mathcal{A}_i$.

An isolated $\mathcal{A}_i$ is closed under interaction. Note that any $X \in \mathcal{A}_i$ is a generator of $\mathcal{A}_i$ under $\mathbf{M}$. A seeding set (section 5.4) of $\mathcal{A}_i$, however, requires at least i *contiguous* objects (along the ith diagonal). The center $\mathcal{C}_i$ of $\mathcal{A}_i$ consists of its i smallest objects: $\mathcal{C}_i = \{A_{i+l,1+l}, \ l = 0, 1, \ldots, i-1\}$ Families can also seed one another cooperatively. For example, to seed both $\mathcal{A}_i$ and $\mathcal{A}_j$ with $i > j$ and i not a multiple of j only one object type from $\mathcal{A}_i$ and one object type from $\mathcal{A}_j$ is needed.

While any single individual object in $\mathcal{A}_i$ is a generator of $\mathcal{A}_i$ under $\mathbf{M}$, it yields a limit cycle with period i under the replacement map $\mathbf{m}$. For the repeated action of $A_{i+l,1+l}$ on itself will follow equation (35), and finally yield $A_{i,1}$. But $A_{i,1} \circ A_{i,1} = A_{2i-1,i}$, so that after exactly i collisions $A_{i,1}$ is seen again.

In a multi-family setting, as occured in our reactor, the action of an object is not restricted to its family. The overall organization, however, does not enable the creation of new families, see (35, 36).

The families, $\mathcal{A}_i$, that happen to be present in the system depend upon its history. For example, suppose that an object of the form $\lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.(x_1)x_3$ was present at an earlier time. Its action on $\lambda x_1.\lambda x_2.x_1 \in \mathcal{A}_2$ yields a member of a new family: $\lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.\lambda x_5.x_3 \in \mathcal{A}_3$. In the reactor the restrictions imposed by the replacement map $\mathbf{m}$ are relaxed due to asynchronous collision and flow events. The production of one single object of a new family can, therefore, be enough to seed that family. Once created, different self-maintaining families are kinetically neutral, and in the long run random exclusion takes place. If the overall organizational structure has reduced to a single family, that family is indestructible since it constitutes an absorbing barrier of type ii. Even if every object species but one, and no matter which one, disappeared, it would suffice to repair the entire family.

Kinetics

Recall that all "reactions" in the model occur with the same unit rate constant. Differential kinetics occurs solely because of network structure, that is, because different objects have varying numbers of incoming pathways that connect to objects with different concentration profiles. While we will not report extensively on kinetics here, we wish to illustrate that allocation of differential rate constants changes the kinetic behavior.

Such a study can be carried out with the ordinary differential equation (ODE) framework (section 5.2). For example, the organization represented by the family $\mathcal{A}_{10}$ was transcribed into the $a^i_{j,k}$ of the ODE system (18). This was done with the 10 smallest object species (the center), and by cutting off all interactions that lead to objects outside the center, thereby artificially closing the system. If all reaction rates are equal (as is the case in the reactor experiments), we find by numerical integration one asymptotically stable fixed point in the concentration simplex, in accordance with our analysis in Stadler *et al.* (1993). This is also the case when reaction rates are assigned randomly and uniformly distributed in $[0, 1]$. It is tempting, however, to distinguish kinetically between the basic cycle,

$$A_{10,1} \circ A_{10,1} \to A_{19,10},\; A_{19,10} \circ A_{19,10} \to A_{18,9},\; \cdots\; A_{11,2} \circ A_{11,2} \to A_{10,1} \tag{41}$$

and all other crossreactions. When assigning rate constants to the cycle (41) that are larger by a ratio of 5 to 1 compared to all other reactions, the dynamics changes into a limit cycle. Thus, organizations are expected to exhibit a range of dynamical behaviors when constructive interactions occur on different time scales.

6.2.3 Example 2

While most of the features characteristic of L1 organizations are present in Example 1, a diversity of other organizations can be obtained by applying different syntactical boundary conditions. The procedure is to rerun the system, while disabling the emergence of a particular organization by declaring as elastic those collisions that produce objects with a grammatical structure characteristic for that organization. Our second example introduces a simple organization that consists of two syntactically different families that depend on one another. It also illustrates how different descriptions are applicable within the same organization.

This example resulted with application as the collision rule, copy actions barred, and syntactical filters, implemented in the form of regular expression filters, that ban objects containing a block of three consecutive abstractions, $\lambda x_i.\lambda x_{i+1}.\lambda x_{i+2}.$, therefore eliminating, among others, the organization of example 1.

The new organization, $\mathcal{O}_2$, consists syntactically of two families $\mathcal{A}$ and $\mathcal{B}$ whose members have the invariant form:

$$A_i \equiv \lambda x_1.(x_1)\lambda x_2.(x_2)\cdots\lambda x_{i+3}.\lambda x_{i+4}.(x_{i+4})\lambda x_{i+5}.(x_{i+5})x_{i+3}, \tag{42}$$

where $i \in \{-2, -1, 0, 1, 2, \ldots\}$ for family $\mathcal{A}$, and

$$B_i \equiv \lambda x_1.(x_1)\lambda x_2.(x_2)\cdots\lambda x_{i+1}.x_{i+1}, \tag{43}$$

where $i \in \{0, 1, 2, \ldots\}$, for family $\mathcal{B}$.

A compressed representation without λ-notation is made through the following correspondences.

$$
\begin{aligned}
a &\equiv \lambda x_i.(x_i) && (44)\\
A &\equiv \lambda x_j.\lambda x_{j+1}.(x_{j+1})\lambda x_{j+2}.(x_{j+2})x_j \equiv \lambda x_j.aax_j && (45)\\
B &\equiv \lambda x_k.x_k\ . && (46)
\end{aligned}
$$

With this compression, family $\mathcal{A}$ can be described as the set of all linear objects consisting of any number of a's terminated by one A. Similarly, $\mathcal{B}$ is the set of objects made of any number of a's terminated by one B. This is a formal language over the alphabet $\{s, a, A, B\}$ with starting symbol s, and the regular grammar:

$$
\begin{aligned}
s &\longrightarrow as && (47)\\
s &\longrightarrow A && (48)\\
s &\longrightarrow B. && (49)
\end{aligned}
$$

An observer could well parse the objects precisely into building blocks of type a, A, and B, without reference to the underlying λ-calculus. This enables the observer to recognize the structure of the system.

All relationships established by the actions of such objects upon one another are described by three laws:

$$
\begin{aligned}
av_1 \circ v_2 &= v_2 \circ v_1 && (50)\\
A \circ v &= aav && (51)\\
B \circ v &= v, && (52)
\end{aligned}
$$

where v, v_1, and v_2 are arbitrary objects of $\mathcal{O}_2$. Note that equation (52) is not observed directly, because of the no-copy constraint. An observer must, however, infer it (see example 53) when attempting a complete description of the invariant subspace.

As was the case in Example 1, the laws allow the prediction of every collision product. For example, the action of aB on $aaaA$ yields aaA according to

$$aB \circ aaaA \xrightarrow{(50)} aaaA \circ B \xrightarrow{(50)} B \circ aaA \xrightarrow{(52)} aaA. \qquad (53)$$

Let us for the moment turn on copy actions again. This only affects the frequency distribution of objects, but not the organizational structure, since an identity action, (52), is implicit in the system's description. In that case, the following description can be obtained.

$$\begin{aligned}
\forall\, f, g \in \mathcal{O}_2 &\qquad f \circ g \in \mathcal{O}_2 && (54)\\
\forall\, f, g, h \in \mathcal{O}_2 &\qquad f \circ g = (h \circ f) \circ (h \circ g) && (55)\\
\exists!\; g^* \;\; \forall\, f \in \mathcal{O}_2 &\qquad g^* \circ f = f && (56)\\
\forall\, f \in \mathcal{B} &\qquad f \circ f = g^* && (57)\\
\exists!\; h^* \in \mathcal{A} \;\; \forall\, f \in \mathcal{A} &\qquad f \circ f = h^*, && (58)
\end{aligned}$$

where $g^* \equiv B$ and $h^* \equiv aaA$. This is a coarse-grained description, because, in contrast to (50)–(52), it does not permit deduction of every interaction product. This description is similar to a group, except that associativity is replaced by the congruence relation (55). Application is not an associative operation (although it could become associative when restricted to some particular subspace).

The reason for this structure is immediate, when we note that the elements A_i and B_i of both families behave as if they were a representation of the natural numbers within the world of Λ. The series $A_0, A_1, A_2, \ldots$ and $B_0, B_1, B_2, \ldots$ correspond to the numbers $0, 1, 2, \ldots$. It also turns out that in what follows A_{-2} and A_{-1} behave like the negative integers -2 and -1, respectively. We will, therefore, abbreviate $\mathbf{i} \equiv A_i \in \mathcal{A}$ or $\mathbf{i} \equiv B_i \in \mathcal{B}$.

The syntactical structure of these numerals is such that the application of one numeral to another performs a difference operation (whose mechanism is readily seen in (50)), although not quite a symmetric one. The full actions are as follows. Within families:

$$\begin{aligned}
\forall\, \mathbf{i}, \mathbf{j} &\qquad (\mathbf{i}, \mathbf{j}) \in \mathcal{A} \times \mathcal{A} \Rightarrow \mathbf{k} \in \mathcal{A}\\
\forall\, \mathbf{i}, \mathbf{j} &\qquad (\mathbf{i}, \mathbf{j}) \in \mathcal{B} \times \mathcal{B} \Rightarrow \mathbf{k} \in \mathcal{B}\\
\mathbf{i} \circ \mathbf{j} \longrightarrow &\quad \mathbf{k} \equiv \begin{cases} j - i & \text{if } i \leq j \\ i - j - 1 & \text{otherwise} \end{cases} && (59)
\end{aligned}$$

Across families in one direction:

$$\begin{aligned}
\forall\, \mathbf{i} \leq \mathbf{j} - \mathbf{2} &\qquad (\mathbf{i}, \mathbf{j}) \in \mathcal{A} \times \mathcal{B} \Rightarrow \mathbf{k} \in \mathcal{B}\\
\forall\, \mathbf{i} > \mathbf{j} - \mathbf{2} &\qquad (\mathbf{i}, \mathbf{j}) \in \mathcal{A} \times \mathcal{B} \Rightarrow \mathbf{k} \in \mathcal{A}\\
\mathbf{i} \circ \mathbf{j} \longrightarrow &\quad \mathbf{k} \equiv \begin{cases} j - i & \text{if } i \leq j - 2 \\ i - j - 1 & \text{otherwise} \end{cases} && (60)
\end{aligned}$$

Across families in the other direction:

$$\begin{aligned}
\forall\, \mathbf{i} \leq \mathbf{j} + \mathbf{2} &\qquad (\mathbf{i}, \mathbf{j}) \in \mathcal{B} \times \mathcal{A} \Rightarrow \mathbf{k} \in \mathcal{A}\\
\forall\, \mathbf{i} > \mathbf{j} + \mathbf{2} &\qquad (\mathbf{i}, \mathbf{j}) \in \mathcal{B} \times \mathcal{A} \Rightarrow \mathbf{k} \in \mathcal{B}\\
\mathbf{i} \circ \mathbf{j} \longrightarrow &\quad \mathbf{k} \equiv \begin{cases} j - i & \text{if } i \leq j + 2 \\ i - j - 1 & \text{otherwise} \end{cases} && (61)
\end{aligned}$$

It is seen that predecessor and successor function are simply given by $\mathbf{1} \in \mathcal{A}$ and $-\mathbf{1} \in \mathcal{A}$, respectively. The structure (54–58) is now immediate. Syntactical closure is evident, $\mathbf{0} \in \mathcal{B}$ is the neutral element everywhere, $\mathbf{i} \geq \mathbf{0}$ (for $\mathcal{B}$) or $\mathbf{i} \geq -\mathbf{2}$ (for $\mathcal{A}$) is its own inverse, and the difference between any two natural numbers is invariant to the subtraction or the addition of the same constant, see (55).

The system $\mathcal{O}_2$ is self-maintaining, in the sense of equation (38), as soon as it contains the predecessor and successor functions. The reason why both families coexist is that they depend on each other. $\mathcal{B}$ lacks a successor function, but $-\mathbf{1} \in \mathcal{A}$ acts as such for both families. On the other hand, $-\mathbf{1}$ cannot be produced within $\mathcal{A}$ alone, but requires interactions with family $\mathcal{B}$.

Any set that contains at least one element $\mathbf{i}$ from $\mathcal{A}$ and one element $\mathbf{j}$ from $\mathcal{B} - \mathbf{0}$, satisfying $\mathbf{i} < \mathbf{j}$, is a seeding set of the organization. The center of the organization is $\{-\mathbf{2} \in \mathcal{A}, \mathbf{0} \in \mathcal{A}, \mathbf{0} \in \mathcal{B}, \mathbf{2} \in \mathcal{B}\}$. Every object is a generator.

6.2.4 Example 3

The third example serves to illustrate that it may take many laws to completely specify an organization. For this experiment the syntactical boundary conditions are set to disallow objects with three consecutive abstractions (as in example 2), as well as patterns of the kind $\lambda x_i.(x_j)\lambda x_k.(x_l)$. The collision rule remains application, and copy actions are barred.

The organization, $\mathcal{O}_3$, resulting under these conditions consists of objects that can be parsed into the following building blocks:

$$\begin{aligned}
a &= \lambda x_i.(((((x_i)\lambda x_{i+1}.x_{i+1})\lambda x_{i+2}.\lambda x_{i+3}.x_{i+2})x_i)x_i) \\
b &= \lambda x_j.(x_j) \\
c &= \lambda x_k. \\
A &= \lambda x_l.ax_l \\
B &= \lambda x_m.bx_m \\
C &= \lambda x_n.\lambda x_{n+1}.x_n \\
D &= \lambda x_o.x_o
\end{aligned}$$

The set of objects constituting the invariant subspace of this organization is specified by a grammar that generates objects consisting of any number of symbols from $\{a, b, c\}$ terminated by one from $\{A, B, C, D\}$, with the constraint that no two bs or cs appear consecutively. Note that the latter is a consequence of the boundary conditions.

The complete set of laws describing all relationships in $\mathcal{O}_3$ is composed of 20 equations.

$$\begin{array}{rcl@{\qquad}rcl}
av_1 \circ aav_2 & = & aav_2 \circ v_1 & av_1 \circ A & = & av_1 \\
av_1 \circ bav_2 & = & bav_2 \circ v_1 & av_1 \circ B & = & bv_1 \\
av_1 \circ abv_2 & = & (v_2 \circ abv_2) \circ v_1 & av_1 \circ C & = & C \\
av_1 \circ cav_2 & = & cav_2 \circ v_1 & av_1 \circ D & = & v_1 \\
av_1 \circ cbv_2 & = & (v_2 \circ cbv_2) \circ v_1 & bv_1 \circ v_2 & = & v_2 \circ v_1 \\
av_1 \circ aB & = & aB & cv_1 \circ v_2 & = & v_1 \\
av_1 \circ bB & = & bv_1 & A \circ v_1 & = & av_1 \\
av_1 \circ cD & = & D & B \circ v_1 & = & bv_1 \\
av_1 \circ aD & = & aD \circ v_1 & C \circ v_1 & = & cv_1 \\
av_1 \circ bD & = & bD \circ v_1 & D \circ v_1 & = & v_1
\end{array} \tag{62}$$

where v_1 and v_2 are arbitrary elements of $\mathcal{O}_3$. The center is the set of constructors $\{A, B, C, D\}$.

The laws express statements similar to those encountered in the previous examples. The system is a network of transformations among the building blocks a, b, c, which are introduced by the constructors A, B, C, respectively. From (62) it is apparent that once the constructors have appeared in the system, they are maintained in it through pathways spawned by their direct or indirect products. It is our observation that the system also persists kinetically.

6.2.5 Example 4

This example is representative of a great many instances where a complete specification of the grammatical structure and of the laws pertaining to the invariant subspace becomes increasingly hard to discern. The last example, $\mathcal{O}_4$, was obtained from random initial conditions, with copy actions barred, and with syntactical filters that prevent the emergence of any of the previous examples.

There are two basic objects that spawn the organization:

$$A \equiv \lambda x_1.((x_1)\lambda x_2.\lambda x_3.x_1)\lambda x_4.\lambda x_5.x_4 \tag{63}$$

$$B \equiv \lambda x_1.\lambda x_2.((x_2)\lambda x_3.((x_1)\lambda x_4.x_2)\lambda x_5.\lambda x_6.x_5)x_1 \tag{64}$$

We further compress the λ-notation with

$$\begin{array}{rcl@{\qquad}rcl}
c & \equiv & \lambda x_j & & & \\
d & \equiv & (x_i) & [& \equiv & (\\
e & \equiv & \lambda x_k.\lambda x_{k+1}.x_k &] & \equiv &)
\end{array}$$

where we have renamed the parentheses to avoid confusion between the notation of the compressed description and the original λ-notation.

A picture of the syntactical structure of the members of $\mathcal{O}_4$ is best conveyed by illustrating selected laws that govern actions of B.

$$\begin{aligned}
B \circ c[v_1]v_2 &= c[dc[dv_2]e]c[v_1]v_2 && (65)\\
B \circ cccv_1 &= c[dccv_1]cccv_1 && (66)\\
B \circ cc[dc[dv_1]e]v_2 &= c[dccv_1]cc[dc[dv_1]e]v_2 && (67)\\
B \circ cc[dv_1]v_2 &= c[dcv_1]cc[dv_1]v_2 && (68)
\end{aligned}$$

where, as before, v_1 and v_2 denote arbitrary objects of $\mathcal{O}_4$. The first law, for example, states that the action of B on any object, with one single c preceding the outermost parentheses, consists in (i) finding the subexpression v_2 enclosed by the outermost pair of parentheses, (ii) sandwiching v_2 between $c[dc[d$ and $]e]$, and (iii) finally attaching to this construct a copy of the original object. Similar procedures are described by the other laws.

In $\mathcal{O}_4$ objects are a variety of nonlinear nestings of other objects of the same organization, down to the two basic objects A and B. Part of the action of A consists, for example, in extracting the v_2 portion from objects produced by B on the right hand side in (65), and prefixing that portion with c: cv_2.

A and B mutually maintain themselves. For example, cB transforms A into B, and cA does the reverse. cB is created from B by action of A, while cA is similarly produced by action of B on A (through an intermediate). A, B, cA, cB is a small self-sustaining ensemble. The entire suite of actions involving all objects that can be constructed from this quartet is very complicated. Most, but not all, actions of B are captured by the regularities (65)–(68), but there is more subtle syntactical diversity in the system than we feel is appropriate to describe.

Despite its complexity, $\mathcal{O}_4$ nevertheless exhibits an invariant self-maintaining syntactical and algebraic structure that can, in principle, be concisely described without referring to λ-calculus.

6.2.6 Summary of Level 1

The examples show that initially unstructured collections of random objects that are functions generate, under constrained flow-reactor conditions and the reaction scheme (15), sets of objects that occupy an invariant subspace of Λ characterized by three properties:

- **Grammar**. The syntactical structure of all objects of the subspace is characterized by a finite number of rules, i.e., a grammar. The subspace represents a formal language, whose grammatical structure is invariant with respect to the applicative interaction of its members.

- **Algebraic structure**. All relationships of action between objects of the subspace are characterized by a finite set of equations. Neither the formulation nor the discovery of these laws require knowledge of the underlying λ-calculus, thus defining an independent level of description. The system often admits a coarse-grained description that emphasizes symmetries and particular roles of objects. The subspace is the free system of its constructors. The constructors are the objects denoted by capital letters in the previous examples. So far they have mostly - but not always - been closed λ-expressions, also known as combinators (Curry and Feys, 1958; Curry *et al.*, 1972; Revesz, 1988; Hindley and Seldin, 1986). To date, the constructors have always been members of the center.

- **Self-maintenance**. There is a subset of the invariant subspace that is realized under flow-reactor conditions, and that maintains itself in the sense of (38). It is a summary of our observations that this subset always contains the center (39), that the center contains the constructors, and that there are no disjoint self-maintaining subsets. Systems that enter an invariant subspace, move within that subspace to the center. The size of the subset maintained in the reactor depends on its capacity. Self-maintenance, however, is a statement about the constructive capacities of a set of objects, and is thus only a necessary but not sufficient condition for kinetic persistence of an organization (i.e., organizations can "die").

These properties are generic for L1-ensembles as they arise in our system. *We take these three properties of object grammar, algebraic structure of action between objects, and self-maintenance as defining a notion of organization.* A major point is that L1-organizations are also *objects*, though objects of a different class than L0-objects. The three closures - syntactic, algebraic, kinetic - induce a *non-spatial* notion of boundary and, therefore, of identity. L1-organizations are unitary entities that can be combined and "applied" to one another as will be seen in section 6.4.

That L1 organizations also qualify dynamically as objects may be illustrated by examination of the equations used to model the dynamics of L0 objects. Consider a system that can be properly described by equation (18). Further assume the no-copy rule, and let the system contain a variety of self-maintaining organizations I such that $\forall I, J\ I \cap J = \emptyset\ \wedge\ I \circ J \subseteq J$. Equation (18) can then be rewritten by regrouping the individual L0-objects according to their L1-membership (Stadler *et al.*, 1993):

$$\dot{x}_{k\in K} = \sum_{i\in K} \sum_{J} \sum_{j\in J} \alpha^{k\in K}_{i\in K, j\in J} x_{i\in K} x_{j\in J} - x_{k\in K}\Omega(t). \tag{69}$$

Now let $x_K = \sum_{k\in K} x_{k\in K}$ and $y_{k\in K} = x_{k\in K}/x_K$, and equation (69) can be restated in terms of the self-maintaining organizations I, at the expense of time dependent coefficients:

$$\dot{x}_K = x_K \left[\sum_J \mathcal{A}_{KJ}(t) x_J - \Omega(t) \right], \tag{70}$$

where the flow $\Omega(t)$ is rewritten accordingly, and with

$$\mathcal{A}_{KJ}(t) = \sum_{i \in K} \sum_{j \in J} \sum_{k \in K} \alpha^{k \in K}_{i \in K, j \in J} y_{i \in K} y_{j \in J}.$$

Equation (70) has the form of the replicator equation (21). Thus, L1 organizations are governed by the same dynamics as are other units-of-evolution.

In additional to the defining features outlined above, we have explored the response of L1 organizations to various forms of perturbation.

Robustness towards copy-actions. The algebraic structure of an L1-organization may contain copy-actions, as exemplified by the necessity of including neutral elements in the description of an organization (e.g., Example 2). Under no-copy conditions these actions are rejected when they result from a collision. Re-allowing copy-actions after an organization has formed leaves the organization unchanged.

Response to relaxing the reaction scheme. Equation (15), the reaction scheme, can be read as the balance equation of two simultaneous events. (i) A and B are used up in the process of producing a new particle, the normal form of $((\Phi)A)B$. (ii) One copy of each, A and B, is supplied from an external reservoir whenever (i) happens. This is, of course, not a buffering of A and B, since their concentrations will change over time due to (a) their removal by a dilution flux, and (b) multiple production channels. Note that this external reservoir implicitly ranges over the closure (with regard to $\circ_\Phi$) of the initial set of objects (see section 5.4).

The necessity of (15) derives from our initial conditions which consist of the maximum possible diversity where each object species is present in a single instance (G. Wagner, pers. comm.). Destroying the only instance of an object upon interaction raises a conflict with the potential necessity for its later presence. This serious concurrency problem can be circumvented by starting out with multiple instances of each object species. We cannot, however, decide *a priori* how many objects are sufficient. Any choice may influence what organizations can arise. Our reaction scheme removes this difficulty. Once self-maintaining closure has occurred (if it occurs), the organization automatically guarantees an adequate number of instances for each object species. At this point (15) can be relaxed to a catalytic transformation as in (16). We find, however, that the organization is less vulnerable to stochastic fluctuations if the original reaction scheme (15) is retained for members of the center, while being relaxed to (16) for all other objects.

Response to perturbations. While a quantification of the behavior towards functional perturbations is not pursued in detail here, we have explored the effects of introducing a small number of random objects (typically three objects in 10 copies each), periodically or as a singular event. In most cases the organization is not altered, because the perturbing agents and their products are not maintained, either constructively or kinetically, and are eventually diluted out. In some instances the perturbing object has spawned interactions that resulted in a stable extension to the previous set of objects. Objects that extend an organization are (by definition) grammatically distinct from objects constituting the organization prior to perturbation. Their insertion may, therefore, require a recasting of the organization's grammatical and algebraic description. In simple instances, such as example 2 (section 6.2.3) and example 3 (section 6.2.4), *all* laws remained valid even in the presence of the new objects. Other laws that take into account the new syntactical building blocks were simply added. Small perturbations have never been observed to destroy an organization, but when they were repeatedly successful in extending a system, we observed the displacement of previous extensions.

6.3 Reproduction versus self-maintenance - the transition from Level 0 to Level 1

L1-organizations are self-maintaining, but not self-reproducing. There is no sense in which one can identify multiple instances of an L1 organization in the current system. This contrasts fundamentally with the case of L0 ensembles, where individual objects are involved in copy reactions and hence are reproducing entities. The distinction between self-maintaining organizations and reproducing objects defines the role of selection in our flow reactor and necessitates the prohibition of copy reactions in our L1 experiments.

Darwinian selection, by definition, acts upon entities capable of reproduction (Vrba and Gould, 1986). When the dilution flux removes an object in our L0 system, selection has occurred, whereas when an object is removed from a L1 organization, there is no selection (at the L1 level)[3]. Thus, selection at the L1 level cannot be invoked to *interpret* the dynamics of equation (70).

Disallowing copy-actions, as we do in L1-experiments, turns off selection in a

[3]Natural selection is often confused with sorting (Vrba and Gould, 1986). Readers having difficulty with this distinction may find it fruitful to ask themselves whether removal of a molecule in the citric acid cycle constitutes selection. Clearly it does not, and neither does the removal of an object within a L1 organization. For Darwinian selection to operate in an L1 experiment, the dilution flux would have to remove entire organizations. This, however, is impossible because these organizations are non-reproducing and hence do not exist in multiple instantiations to remove.

radical way. Since we know that once an organization is established, copy actions can be switched on again, typically without affecting the system's stability (see section 6.2.6), the disadvantage of self-maintaining organizations versus individual replicators is restricted to the inital stages of formation of an organization. The basis of this finding is not mysterious. Since the system is constrained to maintain a constant size, objects that replicate eliminate objects that do not. However, once an organization has been generated, objects that do not reproduce are nonetheless created via transformation pathways involving other non-reproducing objects and, hence, can offset the inherent replicatory advantage of copiers.

The prohibition on copy reactions has the additional advantage of illustrating that organizational grades are achieved in the absence of selection. This restriction, however, may well be criticized as being abiological (P. Schuster, pers. comm.). While a number of authors contend that self-maintaining metabolism preceded self-reproduction (Oparin, 1924; Fox and Dose, 1977; Dyson, 1985; Morowitz, 1992), others hold that self-replication was achieved first (Haldane, 1954; Miller and Orgel, 1974; Eigen and Schuster, 1979). If the latter were the case, it might indeed be regarded as abiological to claim that reproduction was suspended. Hence the restriction on copy reactions in our L1-experiments may be regarded as inappropriate (but see Buss (1994) for a plausible scenario under which this restriction might have occurred even if self-replication preceded the evolution of metabolism).

Accordingly, we have sought to explore further the transition between Level 0 copiers and Level 1 organizations. Returning to our flow reactor seeded with a set of random objects, a race is evident. The time required for a self-maintaining metabolism to dominate the system is longer than the time required for self-replicators to dominate the system. It follows that a transition from self-replication to self-maintenance can be generated either by a process which reduces the rate at which self-replication occurs, or increases the speed with which organizations dominate, or both. We have explored three sets of conditions, motivated by chemical and biological intuition, each of which permits this transition to occur.

Varying copy efficiency

In actual chemical systems that replicate, the process of replication is a complex one and is notably less rapid than simple catalytic reactions. In our system all reactions occur with the same rate constant. A reduced, rather than zero, reaction rate for copy functions should facilitate the rise of organizations (J. Kephart, pers. comm.; P. Schuster, pers. comm.). We have systematically varied the probability with which we accept the product of any copy action in the system under a diversity of initial conditions. We found that reducing the probability of copying to 0.75 was typically sufficient to permit organizations to develop.

The abundance of copy reactions

A naive one-to-one correspondence between λ-objects and actual molecules is not a proper interpretation of our model. In particular, it is far simpler to construct a λ-expression with a copy functionality than it appears to be to design a molecule that replicates (but see, von Kiedrowski, 1986; Tjiuikaua *et al.*, 1990). It is appropriate, then, to explore the consequences of depressing the abundance of objects with copy functionality in our system. If complicated objects, incapable of copying, are used to initialize the system, copy functions may be expected to take longer to arise granting the time necessary for organizations to form first and preempt the available space in the system.

We have sought to explore this possibility by initializing the reactor with substantially more complex λ-expressions than were utilized in earlier experiments. Intricate λ-expressions are unlikely to act as copiers for all objects they encounter in a system. Their copy functionality may become embedded in a network spawned by their own constructive actions which eventually gives rise to an organization. While ensembles of copiers or single self-copiers dominate these systems under some sets of initial conditions, it is not difficult to discover initial conditions under which organizations arise.

Parasites of copy reactions

The replicatory advantage of self-copiers can be reduced if the act of copying sustains non-copying objects. A replicator's rate of reproduction may be effectively controlled by parasites (J. Padgett, pers. comm). In our mean field setting, the build-up of parasites simply dilutes the replicator(s), hence decreasing the frequency of their encounters, and thereby controlling the copy-activity in the system. The following case provides a striking example of this route to organization.

Figure 3 is a rendering of the structure of an organization obtained by prohibiting syntactical structures of the projector kind (L1-example 1), and by allowing objects that are not abstraction-forms to be products of collisions, but barring them from acting upon others.

The organization is centered around a replicator R. There is further a parasite-object P_i that is copied by R. P_i acts in turn upon R and produces an object O_i. O_i is not an abstraction-form, and while our boundary conditions prevent it from acting upon others, others may act upon it. In fact, R acts on O_i and thereby produces P_{i+1}, which is another parasite, syntactically not identical, but

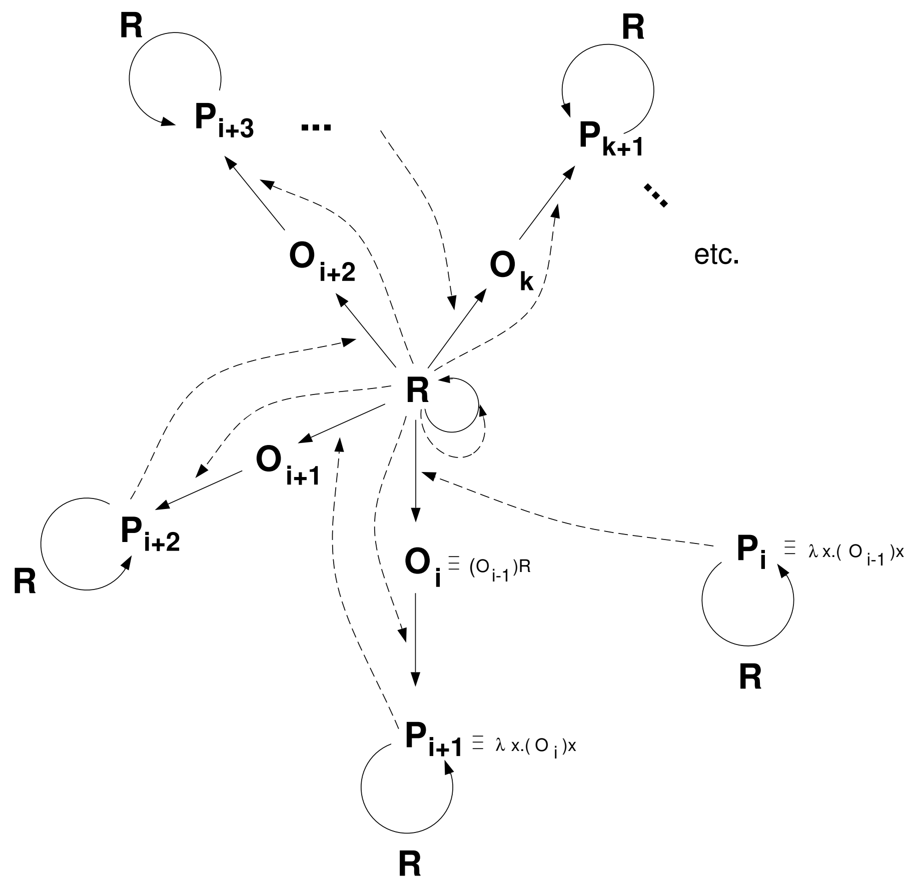

Figure 3: A route to organization through parasites. Solid and dotted arrows are explained in the caption to Figure 1. $R \equiv \lambda x_1.\lambda x_2.(x_1)x_2$. The dots in the upper part of the figure reflect the continuing generation of branches analogous to those shown. See text for details.

analogous in action to P_i. This then leads to P_{i+2}, and so on *ad infinitum* were it not for the upper limit on the size of an object (1000 characters).

Some details are interesting. First note from Figure 3 that at the syntactical level $P_{i+1} \equiv \lambda x.(O_i)x$. While syntactically $P_{i+1} \neq O_i$, in terms of action they would be both equal if O_i were not barred from action, that is: $\forall A \in \Lambda \;\; (P_{i+1})A = (O_i)A$. Hence we may say that functionally $P_{i+1} = O_i$. The P's, therefore, circumvent the constraint that catches the O's; P_{i+1} is literally the "active form" of O_i.

The interactions displayed in Figure 3 produce interesting dynamics. Except for its copying by R, P_i is not produced by anything else in the system at the time illustrated in Figure 3. Suppose (as it does) that a fluctuation wipes out P_i. There is no pathway left to produce P_i, since to copy P_i at least one instance of it is needed. Once P_i disappears, O_i can no longer be produced, and eventually O_i will disappear. This in turn removes an important non-copy pathway to P_{i+1}, making it vulnerable to chance extinction (in particular if the concentration of the P's is low). When this happens (as it does), the story repeats as for P_i. In the stochastic flow reactor each spike originating at R, $R \rightarrow O_k \rightarrow P_{k+1}$, will eventually disappear. At the same time the system provides for the generation of new arms, since the P_k with the largest k will act on R and produce a novel O_{k+1}. The system keeps losing spikes on the low k end, and producing spikes at the high k end. Because of the object size limitation, this process will stop at some point, and the system goes extinct asymptotically. In principle, however, the example illustrates how a set of objects may continue to change indefinitely, while retaining an invariant relational structure.

The recursive structure of the P's and O's (Figure 3) makes them increasingly larger objects. Collisions between the P's are elastic at the time of Figure 3, because their products are larger than the size limitation. The simple structure of the organization in its later phase is, therefore, a result of the boundary conditions. Where did P_i, with the lowest i currently in the system, come from? Following the recursive structure of the P's and O's leads us back to some small putative object O_0 (buried in the structure of all P's and O's), which was generated during the early history of the system.

6.4 Level 2: Organizations of organizations

L1-organizations qualify as an object class (see section 6.2.6), hence it is appropriate to ask how two such organizations might interact. Combining two organizations, $\mathbf{A}$ and $\mathbf{B}$, in a flow reactor will lead to interactions within $\mathbf{A}$, within $\mathbf{B}$, and across organizations. Collisions across organizations need not lead to products that belong to $\mathbf{A}$ or $\mathbf{B}$. Cross-interactions may build up an ensemble of objects that is located, organizationally speaking, "outside" the organizations engaged in the interaction. We denote this ensemble by $\mathcal{C}$. Furthermore, collisions between members of $\mathcal{C}$ may yield products that further expand $\mathcal{C}$ or which belong to $\mathbf{A}$ or $\mathbf{B}$.

Formally, this process is the union of the free products $\mathbf{A} \star \mathbf{B}$ and $\mathbf{B} \star \mathbf{A}$, and $\mathcal{C} = \mathbf{A} \star \mathbf{B} \cup \mathbf{B} \star \mathbf{A} - \mathbf{A} - \mathbf{B}$. Two cases arise: (i) $\mathcal{C} = \emptyset$ or (ii) $\mathcal{C} \neq \emptyset$. The two cases behave quite differently under flow reactor dynamics. In case (i) one of the two organizations displaces the other, depending on the relative magnitude of the transformation flows between them (if any), as well as on their internal growth rate. In case (ii) the set $\mathcal{C}$ may offer a variety of pathways which stabilize the integration of $\mathbf{A}$ and $\mathbf{B}$ into a new metaorganization. We refer to $\mathcal{C}$ as the *glue*, and to the resulting organization as a Level 2 or L2 organization.

6.4.1 Example 1

We begin with an example wherein two L1-organizations that have been generated separately are introduced into the flow reactor with a different collision rule governing the actions involving the glue. Organization **A** is described as example 3 in section 6.2.4. A second L1-organization, **B**, was generated under the same functional and syntactical boundary conditions, but with a differing set of initial conditions.

B is simple, and follows similar principles to those of **A**:

$$\begin{aligned}
e &\equiv \lambda x_i.((x_i)\lambda x_{i+1}.x_{i+1}) \\
b &\equiv \lambda x_j.(x_j) \\
E &\equiv \lambda x_k.ex_k \\
B &\equiv \lambda x_l.bx_l \\
D &\equiv \lambda x_m.x_m
\end{aligned}$$

Notice that organization **B** overlaps with **A** in terms of the syntactical building blocks b, B, and D. The laws are given by:

$$\begin{array}{rclrcl}
E \circ v &=& ev & D \circ v_1 &=& v_1 \\
B \circ v &=& bv & ev_1 \circ B &=& v_1 \circ D \\
ev_1 \circ ev_2 &=& v_2 \circ v_1 & ev_1 \circ E &=& eD \circ v_1 \\
ev_1 \circ bv_2 &=& v_2 \circ v_1 & ev_1 \circ D &=& v_1 \\
bv_1 \circ v_2 &=& v_2 \circ v_1 & & &
\end{array}$$

Objects outside both organizations, i.e., members of the glue, are subject to the same syntactical boundary conditions as the L1-organizations, but interact via the collision rule

$$\begin{aligned}
\Phi &\equiv \lambda f.\lambda g.((\Phi_1)f)(\Phi_2)g \quad \text{with} && (71) \\
\Phi_1 &\equiv \lambda x_1.((x_1)\lambda x_2.x_1)x_1, && (72) \\
\Phi_2 &\equiv \lambda x_1.\lambda x_2.((x_2)x_1)\lambda x_3.x_3. && (73)
\end{aligned}$$

The effect of a collision rule of the form (71) was discussed in section 5.1. Φ_1 and Φ_2 were chosen randomly.

The initial condition was the sum of both L1-organizations as they were populated at the end of the experiments that produced them. Populations **A** and **B** consisted of 54 and 41 different object species, respectively. Each population had 1000 particles. Hence the initial population of the L2-experiment consisted of 2000 particles with 95 different object species. The capacity of the flow reactor was increased to 3000 particles. The L1-organizations overlap with respect to the family of $b \cdots bB$-objects. When such objects are produced by interactions involving glue-objects, they are allocated with equal probability to either L1-organization.

Both **A** and **B** were stably maintained under these conditions. In Figure 4 we show the total number of particles in each organization and in the glue as a function of the number of elapsed collision events. During the initial phase the larger organization **A** grows much faster than **B**. However, the rapidly generated glue stabilizes both organizations, while itself remaining remarkably constant in population size.

The glue contains new objects made of combinations from **A**'s and **B**'s syntactical building blocks. In addition, glue-objects display one new syntactical pattern that behaves like parentheses. We code the new pattern with

$$[\equiv \lambda x.((x) \quad \text{and} \quad] \equiv)D\ . \tag{74}$$

Objects that carry the new pattern are of the form $v_1[v_2]$, where v_1, v_2 are arbitrary objects of the system, and v_1 may be null. The $[v]$-pattern is evidently a consequence of the Φ_2-part (73) of the collision rule.

Algebraic laws can be formulated in terms of application, $\circ$, but typically not in terms of other interactions $\circ_\Phi$. The reason is that application, axiom (3), is a basic constructor of λ-objects. We can, however, always reformulate $\circ_\Phi$-interactions in terms of $\circ$, for the collision rule Φ is itself a λ-object and thus defined in terms of applications. From the previous definition of Φ we obtain:

$$v_1 \circ_\Phi v_2 \equiv ((v_1 \circ cv_1) \circ v_1) \circ [v_2]. \tag{75}$$

With (75) every interaction involving glue-objects can be deduced by using the laws of both L1-organizations together with additional laws expressing new relations established by the glue. Such laws are, for example,

$$\begin{array}{rclrcl} [v_1] \circ v_2 & = & (v_2 \circ v_1) \circ D & av_1 \circ eav_2 & = & av_2 \circ v_1 \\ ev_1 \circ av_2 & = & v_2 \circ v_1 & av_1 \circ ebv_2 & = & (v_2 \circ ebv_2) \circ v_1 \\ av_1 \circ eev_2 & = & (v_2 \circ v_2) \circ v_1 & & \text{etc.} & \end{array} \tag{76}$$

Glue-objects conform with a specific grammatical structure. However, that structure lacks closure since glue-glue interactions may yield members of the constitutent L1-organizations. The glue cannot maintain itself, when separated from the L1-organizations that gave rise to it.

In this example, the two L1-organizations overlapped in components of their syntax. We observed that the overlapping component was lost from one organization in the long run and this same simplification occurred more rapidly if the system was externally perturbed by injections of small quantities of random objects. The simplification, however, does not alter the syntactical and functional description of the L2-organization.

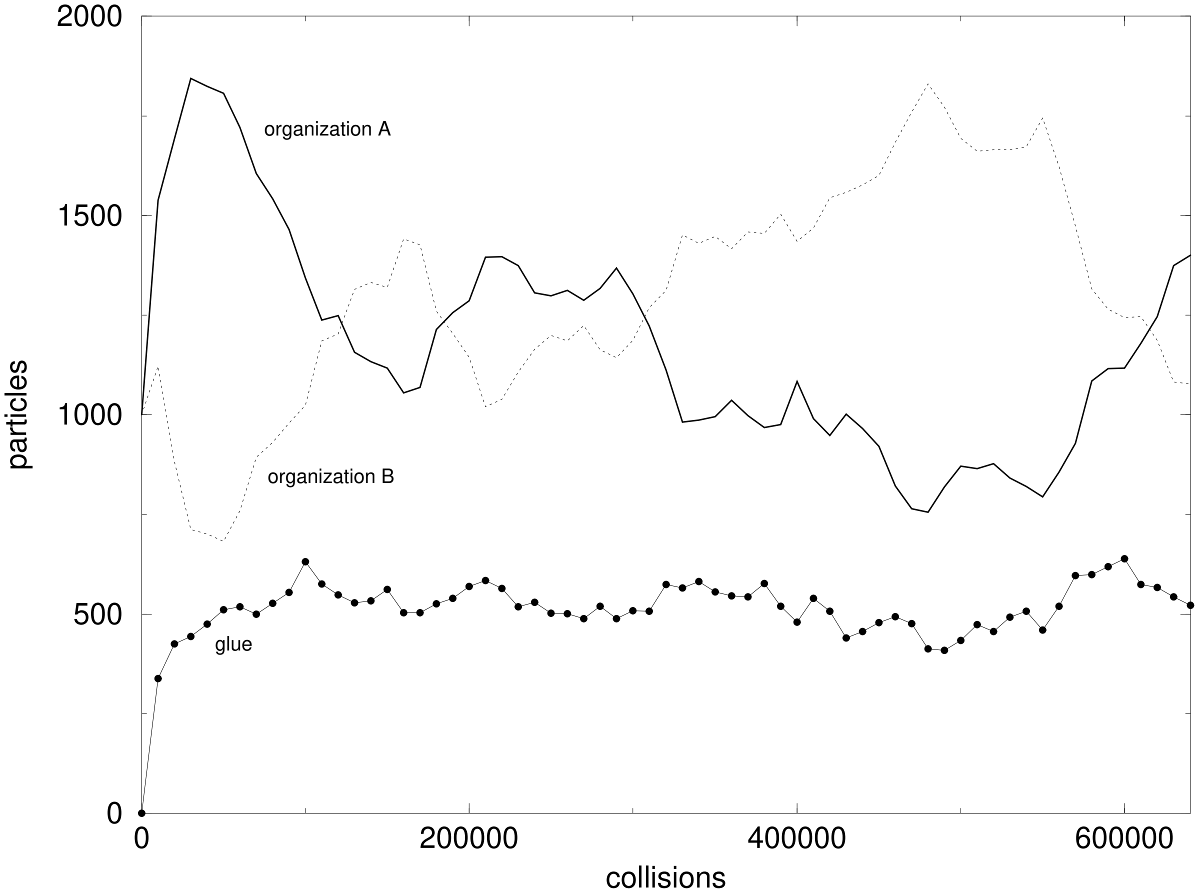

Figure 4: Dynamics of the interaction between two separately generated L1 organizations. The size (number of particles) of each organization and the glue resulting from their interaction is plotted against time (number of collision events). Reactor capacity is 3000 particles. See text for details.

6.4.2 Example 2

In our first example, two L1-organizations were constructed independently and then combined to generate the L2-organization. L2-organizations may also arise spontaneously without external imposition of differing collision laws for either the organizations or the glue. In this example the system was initialized with application as the collision-rule, copy actions barred, and syntactical filters prohibiting objects of the projector form (34) to occur in the initial set or as collision products.

Figure 5 shows the dynamics for both organizations. Organization **A** was easily identified, since it eventually excluded organization **B**. Organization **B** was identified by (i) isolating from the reactor at collision $21 \cdot 10^4$ a set of objects which did not contain syntactical markers associated with **A**, and (ii) inoculating with it an empty flow-reactor. This produced a self-maintaining organization composed of objects whose syntactical structure was resolved to an extent that allowed selective filtering against all other objects present in the original flow-reactor. In addition to the filters for **A** and **B** we formulated a positive third filter for the glue. This allowed us to consistently split the time (i.e., collision) series into **A**, **B**, and their glue. Using this procedure, we accounted for 99.5 to 100% of all particles in the reactor at any time.

Organization **A** consists of objects with the following syntactical components:

$$T \equiv \lambda x_i.\lambda x_{i+1}.\lambda x_{i+2}.\lambda x_{i+3}.(x_{i+3})\lambda x_{i+4}.x_i \tag{77}$$

$$1 \equiv \lambda x_j.(x_j) \tag{78}$$

$$0 \equiv \lambda x_k. \tag{79}$$

Hence objects are made of zero or more 1's separated by one or more 0's and terminated by one T. Stated differently, the invariant subspace is the set of objects described by the regular expression

$$[0] * [1[0]+] * T,$$

where $[pat]*$ and $[pat]+$ denote ≥ 0 and ≥ 1 repetitions of *pat*, respectively. The laws governing the relations among objects of this kind are given by:

$$T \circ v_1 = 0010v_1 \tag{80}$$

$$1v_1 \circ 1v_2 = v_1 \tag{81}$$

$$1v_1 \circ 0v_2 = v_2 \tag{82}$$

$$1v_1 \circ T = 0010v_1 \tag{83}$$

$$0v_1 \circ v_2 = v_1, \tag{84}$$

where v_1, v_2 are arbitrary objects of the subspace. The simplicity of the laws is deceiving. Figure 6 shows a schematic representation of the basic patterns

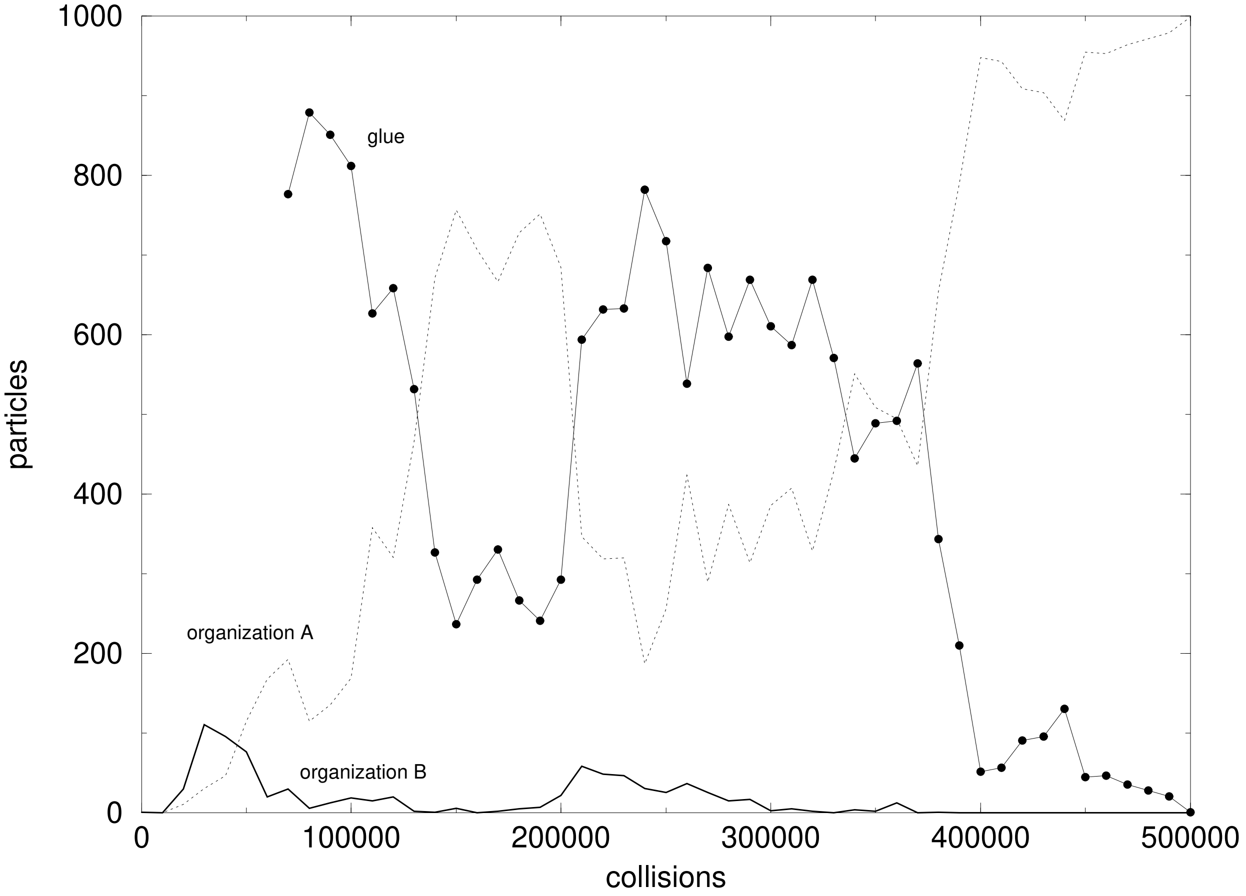

Figure 5: Dynamics of the spontaneous generation of an L2 organization. Two L1 organizations originate spontaneously in the same reactor, and form an L2 organization. The size (number of particles) of each organization and the glue resulting fom their interaction is plotted against time (number of collision events). The glue could be tracked unambigously only after remaining debris from the initial condition had decayed. Reactor capacity is 1000 particles. Typical computer time for such experiments runs from one to ten hours on a MIPS R2000A/R3000 processor. See text for details.

of action, showing mostly self-interactions. A solid arrow represents the transformation operated by the object at the arrow's tail on itself. A dotted line represents a transformation operated by object T. Consider the T in the topmost circle in Figure 6. The entire organization is constructed from T: $T \circ T \to 0010T \circ 0010T \to 010T \circ 010T \to 10T \circ 10T \to T$, which returns us to the starting point. T operating on any object on the circle originates a new circle. For example, $T \circ 10T \to 001010T \circ 001010T \to 01010T \circ 01010T \to 1010T \circ 1010T \to 10T$. Objects on the new circle are origins for further circles, etc. This yields the topmost cluster, the T-cluster, whose objects end with $10T$, except T itself. Now consider the action of $10T$ on T (represented by the bold line labelled with the square): $10T \circ T \to 00100T$, which by iteration eventually again yields $0T$. As before, the action of T will spawn new circles at any point. This builds up the $0T$-cluster, where objects end in $100T$, except for $0T$. By the same token an infinity of clusters can be built, each one identifiable by a base object of the form $0 \ldots 0T$. By virtue of law (84) the repeated self-action of the base objects ends up in T again.

Organization **B** is difficult to understand to the same extent as **A**, by virtue of the fact that the organization sustains a vast number of objects under flow reactor conditions. In isolation **B** sustained 650-800 different object types at any time with a reactor capacity of 1000. Objects of organization **B** were identified as containing the pattern $((x_{i+5})((x_{i+4})x_{i+3})(x_{i+2})(x_{i+1})x_i)$, but neither $((x_i)\lambda x_{i+1}.x_i)$, nor $\lambda x_i.(x_j)$. This was sufficient to discriminate against objects belonging to the glue or to **A**. Most of **B**'s objects are fluctuations present in a single instance, although the organization sustains a small, but well defined and populated core set. Among the core objects are two generators, q and Q:

$$\begin{aligned} Q &= \lambda x_1.\lambda x_2.\lambda x_3.\ \cdots\ \lambda x_8.((x_6)((x_5)x_4)(x_3)(x_2)x_1)\lambda x_9.\lambda x_{10}.x_3 \\ q &= \lambda x_1.\lambda x_2.\lambda x_3.\ \cdots\ \lambda x_8.\lambda x_9((x_7)((x_6)x_5)(x_4)(x_3)x_2)(x_1)\lambda x_{10}.\lambda x_{11}.x_4. \end{aligned}$$

Other well populated objects have the form $\lambda x.\lambda x.\ldots.q$ which produce q after repeated collisions. Self-maintenance is achieved only through interactions between objects in the low populated periphery which directly produce core objects. Yet, the large peripheral set of single-instance-objects is subject to rapid change and turnover. It is likely that this feature ultimately leads to the elimination of **B**.

The diversity of objects characterizing **B** makes a pair interaction analysis or the use of the iterated map (25) infeasible, and the λ-expressions resulting from q and Q proved too complicated for a derivation of the algebraic structure from their behavior.

The glue was syntactically characterized as the set of objects not of the form of **A** but containing $((x_i)\lambda x_{i+1}.x_i)$ or $\lambda x_i.(x_j)$. All objects isolated with the use of these patterns proved to be produced by cross-interactions between **A** and **B**.

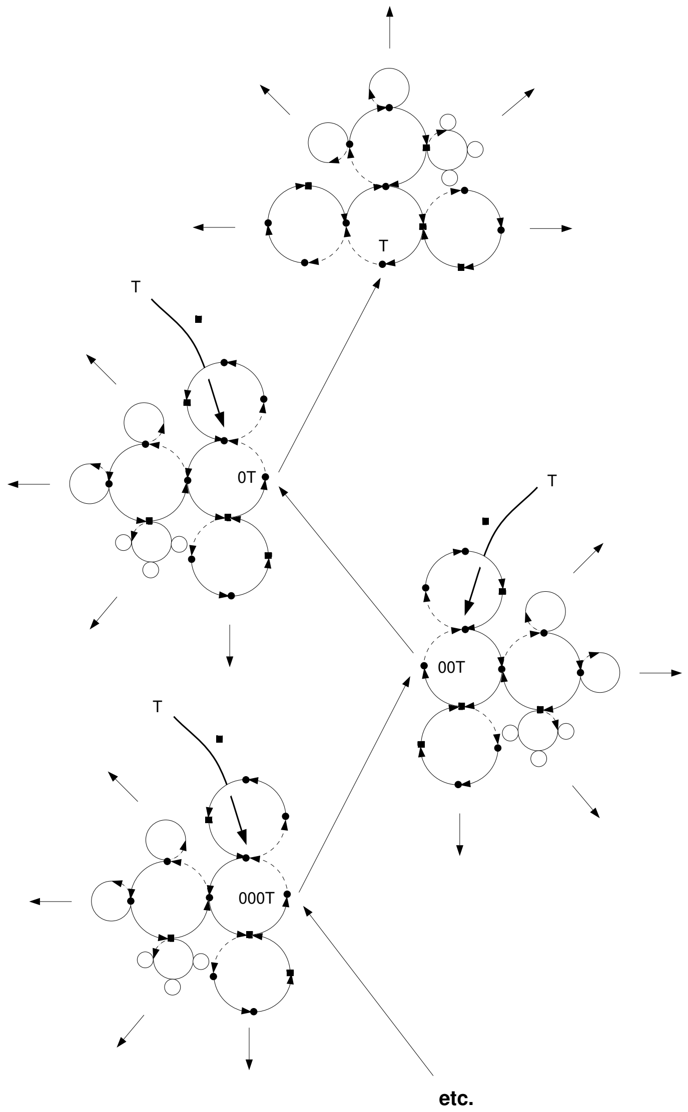

Figure 6: Schematic representation of the relational structure of an L1 organization. Objects are schematically represented as filled dots or filled squares. Squares are objects of the form $10v_1$. Thin solid arrows represent self-interactions of the object "located" at the arrow's tail. Dotted arrows represent transformations operated by object T on the object at the arrow's tail. Thick solid arrows represent transformations operated by "square" objects on T. The size of the circles has no relational interpretation, but indicates the density at which their member objects are maintained in the reactor. This organization is part of an L2 organization. See text for details.

The kinetics shows a series of interesting features (Figure 5). Organization **B** dominates in the early phase of the system, after which organization **A** expands in a burst. Between $12 \cdot 10^4$ and $21 \cdot 10^4$ collisions, when **A**'s population peaks for the first time, the diversity of the system halves from about 620 different object species to slightly above 300. This reflects the fact that **A** is a simpler organization than **B** in terms of its sustained portion under flow-reactor conditions. During the same time **B** completely disappears. At this point the system contains only organization **A** and glue. Nevertheless, **B** is recreated. The glue objects together with **A** retained the capacity to regenerate a seeding set of **B**. This happened repeatedly. At about $20 \cdot 10^4$ collisions **B** returns to a significant extent in the system. The high diversity maintained by **B** is reflected in a burst of the glue size. The glue is the largest component in the system as long as **B** is present beyond some threshold. In the long run **B** is excluded by **A** at about $38 \cdot 10^4$ collisions. As a consequence the glue also disappears.

6.4.3 Summary of Level 2

The preceding examples show that the interaction between L1-organizations can result in a higher-order organization. L2-organizations share with L1-organizations the presence of syntactical, functional and kinetic closure. The major distinguishing feature, however, is the stable *integration* of both L1-organizations as independently self-maintaining subalgebras of the L2-system. That which distinguishes this integration from a coexistence is the presence of a set of objects, the glue, syntactically and functionally "outside" the component L1s.

The center of an L2-organization is exactly the sum of the constituent L1-centers. This allows one to distinguish an L2-situation from a transient where two L1-organizations happen to be present in conjunction with objects which are either remnants of the reactor's prior history or which were introduced exogenously. The proper identification of an L2-organization can be verified by inoculating an empty flow-reactor with the union of the putative L1-centers.

While there is a superposition of L1-centers, it is worth emphasizing that in terms of organizational characteristics an L2-organization is not the sum of the L1-organizations, because the glue which develops is a syntactical and an algebraic extension to both L1 structures. This provides for a fundamentally altered kinetic and constructive situation. The glue on the whole is effectively a catalyst of transformations between L1-organizations which cannot be achieved exclusively within those organizations. At a global level we may write

$$\mathbf{A} \overset{\mathcal{C}}{\rightleftharpoons} \mathbf{B}$$

At the same time the glue is not an organization. It does not preserve its structure when the supporting L1-organizations are removed. Thus, just as Level 1 organizations reflect a new object class, so do Level 2 organizations.

Our experience is that L2-organizations are more readily attained by first generating L1-organizations separately, and subsequently combining them into the same reactor. L2-organizations, however, do not always arise under such conditions. It greatly facilitates the construction of L2-organizations, if the glue, or even the constituent L1-organizations operate under different boundary conditions, in particular different collision rules Φ. moreover, the fact that a glue is produced does not guarantee that both organizations will be stably maintained over long time periods (see example 2, section 6.4.2). Our observations indicate that the kinetic constraints on interactions involving the glue are stringent.

7 Conditions for organizing

The current contribution focusses on introducing abstractions upon which a theory of organization might be based and an implementation within which the consequences of these abstractions might be explored. The results outlined above illustrate that this approach is indeed a powerful tool for generating a diversity of organizations and organizational grades and a tractable device for exploring their properties. Ultimately, however, a theory of organization must provide a suite of formalisms that define what organizations are possible and what variations on such organizations may be achieved. We briefly discuss issues which the current study has identified as central.

Generating an organization involves establishing a network of construction pathways. The formation of a network requires the existence of equivalence classes implied by the reduction process. The size and diversity of equivalence classes implied by a given choice of initial objects will determine the horizon of potential network connectivities.

If the reactor system is not perturbed exogenously, then every object at any time can be expressed as a series of applications involving only objects that were present initially in the system. In other words: any set of objects resulting from an initial set $\mathcal{A}$ after l collision events must be a subset of $\mathcal{S}_{2^l}(\mathcal{A})$, as defined by equation (27), section 5.4.

Consider equation (27), and let us define the following two sets. One set, $\mathcal{S}_n^{NR}$, consists of all expressions obtained by associating in all possible ways up to n applications among objects. This is the set $\mathcal{S}_n$ from equation (27) when no reductions (NR) were executed. Let us refer to an element of $\mathcal{S}_n^{NR}$ as a "collision sequence," because it contains the entire history of collision events all the way back to the initial set. For example, if the initial set $\mathcal{A}$ contains the objects a, b, c, d, then $a \circ ((b \circ d) \circ (c \circ c))$ is a collision sequence, and it appears for the first time as element of $\mathcal{S}_4^{NR}$. The other set is just $\mathcal{S}_n$ as defined in equation (27). This is the set of expressions obtained when all expressions in $\mathcal{S}_n^{NR}$ are reduced.

Let $|\mathcal{A}|$ denote the size of set $\mathcal{A}$, and consider two extreme cases of (degenerate) reduction processes. On the one hand consider the trivial process that establishes each object as its own equivalence class (no reduction), in which case $|\mathcal{S}_n| = |\mathcal{S}_n^{NR}|$ for all n, and no network can ever be formed. The system's diversity explodes like an ever branching tree. (This is the reason for barring λ-expressions that are not abstraction forms from being operators. See section 5.3.) On the other hand consider a reduction process that induces one and only one equivalence class for all objects in the universe (an inconsistent system), in which case $|\mathcal{S}_n| = 1$ for all n, and the network would be trivial, consisting of a single node only. Clearly, a necessary and sufficient condition for organization to occur lies somewhere in between.

Since normal forms represent equivalence classes of objects, we may say that for organization to occur it is necessary that the total number of normal forms, $S_n(\mathcal{A}) = |\mathcal{S}_n(\mathcal{A})|$, grows with lower order than the total number of collision sequences, $S_n^{NR}(\mathcal{A}) = |\mathcal{S}_n^{NR}(\mathcal{A})|$, given an initial set $\mathcal{A}$, of size $a = |\mathcal{A}|$:

$$\lim_{n\to\infty} = \frac{S_n(\mathcal{A})}{S_n^{NR}(\mathcal{A})} = 0. \tag{85}$$

In the following we will omit the reference to the initial set $\mathcal{A}$. Consider s_n as defined in equation (26), and for simplicity let s_n also denote the size of the set s_n. Then $s_{n+1} = S_{n+1} - S_n$, and we replace condition (85) by

$$\lim_{n\to\infty} = \frac{s_n}{s_n^{NR}} = 0, \tag{86}$$

with

$$s_n^{NR}(\mathcal{A}) = \frac{1}{n+1}\begin{pmatrix} 2n \\ n \end{pmatrix} a^{n+1}. \tag{87}$$

$s_n^{NR}(\mathcal{A})$ counts the number of ways in which we can associate n (binary) applications between $n+1$ expressions (Catalan numbers) times the variations allowable by the size of the initial set. Equations in the algebraic structures defining our organizations are statements about the symmetries of collision sequences (associativity, for example) or variations within a collision sequence (commutativity, for example).

While s_n^{NR} always increases exponentially, our examples suggest that s_n increases not faster than polynomially for organizations maintained under flow reactor conditions. For example, consider the subfamily $\mathcal{A}$ of example 2 (section 6.2.3). We build the sets $\mathcal{S}_n$ from an initial set $\mathcal{A}_0$. We can always express set $\mathcal{S}_n$ in terms of the action among (disjoint) difference sets:

$$\mathcal{S}_n = \bigcup_{0 \le i \le n-1} \Delta\mathcal{S}_i \circ \Delta\mathcal{S}_{n-i-1} \;+\; \mathcal{S}_{n-1}, \tag{88}$$

where $\Delta\mathcal{S}_0 \equiv \mathcal{A}_0$ and $\Delta\mathcal{S}_i = \mathcal{S}_i - \mathcal{S}_{i-1},\ i = 1, 2, \ldots$. Hence

$$\Delta\mathcal{S}_n = \bigcup_{0 \le i \le n-1} \Delta\mathcal{S}_i \circ \Delta\mathcal{S}_{n-i-1}. \tag{89}$$

Now recall the interpretation of organization B, equation (59), where the action among objects essentially amounts to a difference operation among the corresponding numbers. Evidently, for $i \le j$, $j - 1 \ge 0$, and for $i > j$, $i - j - 1 \ge 0$. Hence any initial set produces only positive numbers.

Let us start with $\mathcal{A}_0 = \Delta\mathcal{S}_0 = \{-2, -1, 0\}$, and $\Delta\mathcal{S}_1 = \{1, 2\}$ is obtained by direct computation. The largest numbers added to any $\mathcal{S}_n$ will come from (a) the smallest number, $-2 \in \Delta\mathcal{S}_0$, operating on the current largest one in $\Delta\mathcal{S}_{n-1}$ according to $j - i$, and (b) from the reverse situation, where the action takes place according to $i - j - 1$. Let $M = \max\{i \in \Delta\mathcal{S}_{n-1}\}$ for $n > 1$. The largest number in $\Delta\mathcal{S}_n$ produced according to case (a) above is $M + 2$, and the largest number according to case (b) is $M + 1$. It follows that the two largest numbers in $\Delta\mathcal{S}_{n+1}$ must be $M+4$ and $M+3$. Hence by induction the increments $\Delta\mathcal{S}_i$ are contiguous, and contain exactly two elements. $|\Delta\mathcal{S}_n| = 2$ for all n, and $|\mathcal{S}_n| = 3 + 2n$. Clearly, our conjecture regarding polynominal growth and condition (86), or (85), are fulfilled.

Condition (86) is a statement that addresses only network existence, and does not take into account issues of self-maintenance, or any kinetic issues as they arise from network connectivity. A theory of organizations will ultimately require understanding of three issues: (i) what symmetries of interaction sequences are implied by an initial set of objects, (ii) what combinations of such symmetries can achieve self-maintenance, and (iii) how robust are these combinations in regard to changes in the reaction scheme. While our work defines these issues as central, it is apparent that the bulk of the task in generating a theory of organization remains before us.

8 Discussion

8.1 The approach: a minimal chemistry

We claimed at the outset that the formal structure of evolutionary theory is plagued with an existence problem - that it assumes the prior existence of the entities it acts upon. We have introduced a theoretical framework in which we show that self-maintaining organizations can be generated on the basis of two key abstractions from chemistry. Hence, we claim that our approach offers the basis for a constructivist augmentation of the traditional evolutionary approach based on the dynamics of pre-existing entities.

Our approach should be compared with at least three existing research traditions.

Autopoiesis - Lambda-organizations appear to be related to what Maturana and Varela (1973, 1980) have termed "autopoietic systems" (see also Bachmann *et al.*, 1992; Luisi, 1993). From Varela et al. (1974, p.188), "The autopoietic organization is defined as a unity by a network of productions of components which participate recursively in the same network of productions of components which produced these components." In addition to the work by Maturana and Varela, a paper by Rössler (1971) on generalized autocatalytic chemical organizations must be included among prior work to which our model is relevant. No formal morphism between the λ-organizations and the autopoietic tradition or Rössler's line of thought is possible, since these approaches have been exclusively verbal. The λ-system may be considered a formalization of the network component of autopoietic systems and a theory of their origins.

Concurrent computation - Our formulation of organization draws heavily on foundational concepts of computer science. This is not coincidence. The notion of a function as a rule is crucial in achieving the proper abstractions from chemistry. The many-body perspective developed here (and emphasized in Forrest, 1990) is reminiscent of concurrent computation. Concurrency is also at the root of our reaction scheme, equation (15). Once the system has produced an organization which maintains sufficient instances of each object, it has effectively solved part of a concurrency problem by homeostasis, and the reaction scheme can be relaxed.

We recently became aware of work in computer science which also employs a chemical metaphor, albeit in a far less abstract manner (Berry and Boudol, 1990). Their "Chemical Abstract Machine" is proposed as a new paradigm for models of concurrent computation. Similar to our approach here, the machine is a multiset, i.e. a milieu where objects can float freely and interact according to specific reaction rules. In line with Banatre's and LeMetayer's (1986) Γ language the authors pursue a methodology where concurrent programming is "liberated from control management" by having the system's "components move freely and communicate when they come in contact". While this system has not been employed for the study of biological organization, our experience with the λ-system strongly suggests that such an approach would be fruitful.

Autocatalytic reaction networks - Our results replicate a variety of earlier findings by researchers exploring autocatytic reaction networks. Specifically, the capacity of the λ-system to generate ensembles of hypercyclically coupled copying reactions is an independent rediscovery of the results of Eigen and Schuster

(1977, 1978a,b; 1979), as is our finding of the well-known sensitivity of these systems to perturbation in a mean field setting (Neisert *et al.*, 1981). Similarly, the generation of self-maintaining ensembles recalls attempts to construct dynamical models of polypeptide or polynucleotide based metabolisms (Kauffman, 1971, 1986; Eigen and Schuster, 1982; Bagley and Farmer,1992; Bagley *et al*, 1992; Futrelle and Miller, 1992; Thürk, 1993). While the λ-system permits exploration of the logical implications of the features that we have abstracted from chemistry, much of actual chemistry is absent. It is therefore encouraging that the λ-system replicates results obtained using methods grounded in proximity to chemical detail. Note, however, that the tradition of research on autocatalytic reaction networks is distinguished by the absence of a notion of an organization, in the sense defined here.

Our approach contrasts with the autopoietic tradition in having a formal foundation, and differs from both autocatalytic reaction network traditions in the level of chemical detail that our model seeks to capture. We did not attack the problem of organization by an attempt to imitate or emulate the details of actual chemistry. Our model is neither a theoretical chemistry, nor is it a theoretical biophysics. We consider the behavior of a constructive dynamical system, i.e., the many-body phenomena that occur within an ensemble of formal objects that are capable of playing any of three roles: function, argument, and value. We argued that these features are required to realize a minimal "chemistry" in the sense of a constructive system implementing equivalence relations.

The virtues of the minimal approach lie in the realm of representation. Clearly, every organization, natural or artificial, will rely on some representation, since it must consist of objects which have to be expressed in some fashion. Hence there will be features which are imports from the chosen representation, and which consequently are not generalizable to features of abstract organizations. The virtue of our approach based on λ-calculus is to minimize the representational constraints and to keep them transparent. It, therefore, becomes, easy to recognize what is λ-specific and what is not.

It is appropriate, then, to ask what *kind* of biological conclusions can possibly be inferred from such a minimal construction? Since strict analogy between molecules and λ-expressions is inappropriate, only conclusions regarding the features of organizations should be drawn. What is interpretable and observable are (i) the presence of functional and syntactical regularities in organizations, (ii) basic organizational features like closures, self-maintenance, centers, robustness, layered expansions, hierarchical combinations, simplifications, roles or positions within a network structure, and (iii) the linkage between boundary conditions, in particular the functional ones, and the organizational classes to which they give rise. Indeed, these are the very organizational issues claimed at the outset to be at the core of difficulties with evolutionary theory. As we argue in the next section, under biologically motivated boundary conditions, the phenomena arising in this mathematical world share fundamental features with biological organizations and their historical progression.

8.2 The organization problem in evolutionary theory

Given the capacity of the λ-system to rediscover and extend earlier findings, we are encouraged to suggest how the properties of λ-organizations might inform outstanding problems in evolutionary biology. At the outset we claimed that the dual criteria of self-maintenance and self-reproduction defined chemical transformation systems as biological. The latter criterion has received the preponderance of attention in that it is this feature that triggers the Darwinian process. We hold, however, that the former is central to gaining an understanding of (i) the emergence and (ii) the mutability of biological organizations.

8.2.1 The emergence of organizational classes: self-maintenance versus self-reproduction

The history of life is a history of the emergence of new organizational grades and their subsequent diversification (Buss, 1987). A transition in organizational grade occurred when self-replicating molecules gave rise to (or became incorporated within) self-maintaining procaryotic cells and the origin of procaryotes was itself followed by the emergence of a hierarchical nesting of different procaryotic lineages to generate multi-genomic eucaryotic cells. The existence problem in evolution is germane not only to the origin of life, but to a number of transitions in organizational grade. We have found, in λ-organizations, transitions that mimic the transitions seen in the history of life, i.e., the transition from self-replicating molecules to self-maintaining organizations to hierarchical combinations of such organizations.

Level 0: replicator ecology

Level 0 (section (6.1)) consists of a single self-copying object or of an ecology of objects *all* of which are engaged in mutual copy-actions. This is precisely the situation described by the replicator or ecological Lotka-Volterra framework: the reproduction of an object species, A, requires itself and one (or more) other object species, B. From a purely functional point of view A is a "fixed point" of its interaction with B. This is reflected in the factorization for the rate equations in the relative concentrations, $\dot{x}_i = x_i\, F_i(\vec{x})$.

We find that copy ensembles, of which the hypercyle (Eigen and Schuster, 1979) is one example, spontaneously arise but are unstable to perturbation in a mean-field setting. Two kinds of perturbation are relevant here: the exogenous addition of random objects (including copiers), and the random deletion of subsets of the system's support due either to exogenous intervention or to endogenous stochastic fluctuations. It is characteristic of Level 0 ensembles that they are unstable towards both perturbations.

The instability of L0 ensembles to the addition of new objects is in most cases attributable to parasitism by single objects which are copied by members of the ensemble and thereby achieve a replicatory advantage. Identity functions, in particular, play a strong kinetic role, since they efficiently resist the dilution flow by autocatalytic duplication. This has been successfully countered in other formulations by a spatial implementation (Bjoerlist and Hogeweg, 1991).

The instability upon removal of object species, however, reflects an intrinsic lack of self-repair. Because copying is not a constructive operation, the ensemble easily achieves trivial closure. Stated differently, an identity function is not an interesting constructor because of idempotency. Eliminating construction eliminates the capacity for self-repair.

Parasites, side-reactions and organizational transitions

Copiers, i.e., identity functions, come to dominate L0 systems under perturbation preventing the emergence of new organizational grades (e.g., L1 or L2). Pure replicators have two inherent advantages, they reproduce themselves and they maximize this advantage by not entertaining constructive interactions with others. There are, therefore, two routes to a transition in organizational grade: (i) control of the inherent kinetic advantage of replicators, and/or (ii) control of replicators via unavoidable constructive side-reactions.

The first route may be achieved in at least three ways. First, one may prohibit copy interactions in our system, with the result that L1 organizations appear. This condition corresponds to the "metabolism-first" school of the origin of life (Oparin, 1924; Fox and Dose, 1977; Dyson, 1985; Morowitz, 1992) or to an early life scenario in which the life cycle of a protocell has a non-proliferative phase (Buss 1994). Second, a modest reduction in the efficacy of copy reactions results in L1 organizations. Finally, one may mimic the relative rarity of copy reactions in actual chemical systems by using more complicated λ-expressions. These may act as copiers on some objects in the population, but will perform transformations on others. Such side-reactions can be sufficient to construct an organization.

Rather than invoking kinetic controls external to the system, our work also suggests an endogenous route by reducing the kinetic advantage of self-copiers through parasites on those copiers. A copier dilutes itself as soon as it sustains objects which are not engaged in copy-actions (J. Padgett, pers. comm.). If these parasites entertain constructive interactions, a direct route from copy-ensembles to self-maintaining organizations becomes possible. We have observed a variety of such cases. A particularly simple and elegant case has been discussed in section

(6.3).

Our model is silent about the likelihood of each of these routes in real chemistry or in the actual history of life. It serves, however, to illustrate that a replicator-based, "genetics-first" approach to the origin of life (Haldane, 1954; Miller and Orgel, 1974; Eigen and Schuster, 1979) and a "metabolism-first" approach (Oparin, 1924; Fox, 1977; Dyson, 1985; Kauffman, 1986; Morowitz, 1992) are both, in principle, possible. The endogenous mechanism based on the kinetic effect of parasites is a novel proposition.

Level 1: a new self

In contrast to Level 0, Level 1 organizations are infinite invariant subspaces of objects, of which only a specific finite portion is maintained under flow reactor conditions. Hence the set of object species in the reactor is no longer closed under interaction. Nonetheless, the organization endogenously maintains grammatical and functional boundaries (the invariances of the subspace). It is these boundaries which we define as an organization and which qualify the organization as a new object class. Moreover, the organization is self-maintaining and characterized by kinetic persistence, growth from a minimal subset of L0 objects, and robustness to perturbation.

Note, however, that while L1 organizations maintain themselves kinetically and constructively, they do not reproduce. In no sense can one identify multiple instances of the same L1 organization in our flow reactor. While L0 objects are constructed and lost in our system, this is no more a matter of loss or gain of an organization than, say, fluctuations in the number of molecules present in the citric acid cycle of a bacterium imply a change in the number of bacteria. Reproduction at the new object level requires a means for separating two instances of a L1 object.

Hierarchical progression of organizational levels

In addition to the transition from self-reproduction to self-maintaining organizations (i.e., L0 to L1), we have shown that self-maintaining Level 1 organizations can be hierarchically combined to produce new self-maintaining Level 2 organizations. The Level 1 organizations of which these metaorganizations are composed retain their capacity for self-maintenance. The λ-framework, thus, exhibits the capacity to generate the hierachical nesting of units that has so characterized the history of life.

Note that the capacity of the λ-system to mimic transitions which have occurred in the history of life is a consequence of self-maintenance, rather than reproduction. In this study, the only reproducing entities are the self-copying functions arising in Level 0. Level 1 and Level 2 organizations are self-maintaining, but not self-reproducing. Since Darwinian selection presupposes the existence of reproducing entities (Lewontin, 1970; Vrba and Gould, 1986), it follows that these transitions are obtained without appeal to natural selection.

8.2.2 The origin of variation

The famous macro-mutationist Hugo DeVries closed his 1904 book (p. 825-826) with the line "Natural selection may explain the survival of the fittest, but it cannot explain the arrival of the fittest." The issue which prompted this remark was concern that since all novelty must arise from mutation, whereas natural selection merely dispensed with the unfit, that the principle problem in evolution was to understand how mutation could give rise to phenotypic novelty. Selection, he held, has no generative power, hence evolution was largely a problem of what generates the fit.

Biologists today, as then, have no coherent theory of variation. Population genetics has succeeded largely by virtue of a device which permits this problem to be artfully ignored. Here genes are held to be alleles and alleles are recognized by phenotypic characters. In standard parlance, then, genes are equated to phenotype or, if you like, phenotype is reduced to genes. It is widely appreciated that this conceptualization collapses what are fundamentally two different mappings into one. The mapping of genotype to phenotype and the mapping of phenotype onto fitness are both subsumed under one mapping of alleles onto fitness. The origin of the phenotype - DeVries' arrival of the fittest - is swept under the carpet. Clearly, difficulties arise to the extent that the mapping of the genotype to the phenotype placed limits on what is possible - limits that preclude the arrival of the fittest. Critiques of pan-selectionism (Gould and Lewontin, 1979), characterizations of evolution as "tinkering" (Jacob, 1982), and the recent preoccupation with developmental constraints (Maynard Smith, et al., 1985) are but a reawakening of the same concern. Lambda-organizations provide a foundation upon which to assess these issues.

An organizational view of phenotype

A λ-organization has a "phenotype". It is the set of regularities which specify its grammatical and algebraic structure. Our λ-system, however, is not based on details of an actual chemistry, and the question of the relevance of the internal structure of λ-organizations, and hence of the mutability of these structures, to biological organizations is central. Grammatical and functional regularities, though, are typical for biological organizations as well.

Let us first consider grammatical regularities. Biological metabolisms have as components a variety of families consisting of syntactically related molecules, such as polypeptides, polynucleotides, sugars, lipids, and so forth. Even membranes could be viewed as regular structures with a simple grammatical description. It is characteristic for a λ-organization at Level 1 and Level 2 to consist of objects that are described by specific syntactical regularities. Organizations whose sets of objects can be partitioned into multiple families, each described by a different grammar, have been observed as well. For example, Level 1, Example 2 (section (6.2.3) is a simple instance of an organization based on the interaction between two different syntactical families. Level 2 organizations are similarly structured. The reason for such regularities in λ-organizations is the generation of some basic set of constructor objects from which families within an organization are built up by iterated action.

Since structure is related to function, regularities of the former imply regularities of the latter. Thus, sugars and lipids play specific functional roles in a biological organization. Similarly, in a λ-organization, an object or a set of objects can *have* a specific function, be it kinetic or constructive. For example, in Level 1, Example 2 (section 6.2.3), the syntactic family $\mathcal{B}$, from the viewpoint of $\mathcal{A}$, *has* the function of providing objects required for the production of -1, without which the self-maintenance loop could not be closed. It is, thus, apparent that the organizations generated by our model have an internal structure which assigns "functional roles" to certain elements or families of elements[4].

Response to perturbations

Variation in λ-organizations is highly constrained. Specifically, perturbations rarely alter the algebraic laws that characterize a system. The basis for this restriction is that for any novel object to persist, it must generate a set of transformation pathways that (i) produces itself and (ii) does so in a kinetically effective manner *from within an existing network of pathways.* However, when a novel object does become established, the algebraic laws characterizing the organization are necessarily recast. We have not, though, ever observed a case in which perturbation of one organization has given rise to a fundamentally different one. For example, we have never observed the rebuilding of the "projector organization" (example 1, section 6.2.2) into the "number organization" (example 2, section 6.2.3) upon perturbation. The implication is one of a universe of different organizations, each of which can be modified and diversified, but which are not readily transformed one into another.

The resemblence between λ-organizations and biological organizations is further reinforced by consideration of the response to perturbations. Consider the orga-

[4]Note the distinction between an individual object that *is* a function (in the mathematical sense) and an object within an organizational context which may, in addition, *have* a function (in a "semantic" sense). It is not uncommon to find argumentation wherein an organization is claimed to exist because of the function it has. This, however, is circular since the latter cannot be argued without the former. Our model completely escapes teleology by generating organizations, and consequently function in a "semantic" sense, purely as a many-body phenomenon of objects that *are* functions in the former and well defined mathematical sense.

nization's center, that is, the smallest kinetically persistent set of objects which specifies the construction of the organization. It is characteristic for centers to obey a superposition principle in regard to the construction of new organizations upon perturbation, whose extensional characterization, however, does not superpose linearly. Centers compose, organizations not. This is a characteristic feature of λ-organizations. The fact that organizations must be constructed implies a linear behavior at the level of centers and a nonlinear behavior at the level of the organization.

In addition, we find differences in the effects of perturbation on Level 1 and Level 2 organizations. Successful perturbations of Level 1 organizations led to the elaboration of new transformation pathways and a recasting of those laws required to characterize the organization. In contrast, similar perturbation intensities on Level 2 organizations with overlapping L1-centers (example 1, section 6.4.1) resulted not in an augmentation of existing laws, but in the simplification of one of the centers of a constituent Level 1 organization, while maintaining the Level 2 status of the system. These results recall, respectively, the well-known fact that most metabolic pathways of modern organisms have their origins in procaryotes and that the genomes of organelles have often become simplified by the loss of genes whose function is redundant with that encoded in the nucleus.

These results suggest that an understanding of the origin of variation - that which defines what the Darwinian process can attain - must ultimately be based on a theory of biological organization. In particular, assessing the potential of perturbations in transforming one organization into another will require an appropriate notion of *distance between organizations*. In other words, we require not only a theory of organization, but a theory that defines the limits of its mutability.

8.3 "Organization" as a universality class?

The ease with which biological interpretations are possible in our abstract approach raises the possibility that we are confronted with a universality class, i.e., the possibility of a theory of organization that is independent of the details by which actual systems implement, generate, and evolve organizations.

An example of a universality class in mathematics is computability. It denotes a mathematical structure whose theory is independent of the details of implementation, as manifest by the variety of equivalent models of computation. Despite the existence of specialized theories for each of these models, and despite their fundamentally distinct suitability for particular tasks, there is a level of description, recursive function theory in this case, at which they share a common structure.

Suppose, for example, that we had been motivated not by organizational issues in biology, but rather by issues in earth sciences. It is surely appropriate to treat aspects of the global climate system as problems in the generation and maintenance of systems of chemical transformations. With appropriate syntactical constraints, e.g., limiting the capacity for "polymerization", and functional constraints, e.g., multiple collision schemes operating on vastly different temporal scales, the current model would likely prove as relevant to geological organization as it does to biological organization. While the geochemical example is an obvious candidate, functional organization issues are central in a number of disciplines where the chemical metaphor is almost certainly inappropriate (Winograd and Flores, 1986; Lakoff, 1987; Leifer, 1991; Padgett and Ansell, 1993; Lane, 1993a; Lane, in press). It is, of course, impossible to determine if there indeed exists a universality class "organization" without alternative formalizations beyond those of actual chemistry and our λ-system.

We think it useful to attempt an implementation independent mathematical theory of abstract functional organizations which is based on necessary and sufficient assumptions. In our specific application, we have developed a constructive dynamical system, i.e., a dynamical system with an endogenously induced motion of its support. The points on the trajectories of that motion are sets. Intuitively these trajectories have attractors which we have characterized as algebraic structures. We have, however, not yet produced a mathematical theory of the observed phenomena. We lack the ability to define what organizations are possible or the limits to their variation. Thus, our experimental λ-reactor and the theory specific to it are only stepping stones towards a larger goal.

Our progress toward that goal is mirrored in the state of evolutionary theory. Existing evolutionary theory is a formulation of the process of natural selection, but is incomplete in that it assumes the prior existence of selectable units and is formalized without a theory of the origin of variation. The Darwinian theory, thus, requires augmentation with a theory of the organism. A complete theory of biological organization would explicate what organizations can emerge, combine and vary. The λ-system, based as it is on abstractions from chemistry, succeeds as a device to construct self-maintaining organizations and to explore their properties. In its current form, however, it is only a step toward a theory of biological organization. A formal apparatus for defining possible organizations and possible variations must be the ultimate goal.

Acknowledgements

We dedicate this paper to Professor Alonzo Church on the occassion of his 90th birthday. We thank Gyorgy Revesz for providing source code for a fast λ-calculus reduction machine, John McCaskill for discussions that proved central to the early phases of this project, and Neil Blackstone, Francesca Chiaromonte, David

Lane, John Padgett, James Pelkey, John Reinitz, Peter Schuster, Lee Segel, Burt Singer, Peter Stadler, Gabriel Stolzenberg, Andreas Wagner and Günter Wagner for discussions and careful reading of the manuscript. This paper is communication number 7 from the Center for Computational Ecology of the Yale Institute for Biospherics Studies.

9 Literature

Anderson, P.W. 1972. More is different. *Science*, **177**, 393–396.

Baas, N.A. 1993. Emergence and Higher Order Structures. In: *Proceedings of the Conference on Systems Research and Cybernetics*, Baden-Baden. In press.

Bachmann, P.A, P.L. Luisi and J. Lang. 1992. Autocatalytic self-replicating micelles as models for prebiotic structures. *Nature*, **357**, 57-59.

Bagley, R.J., J.D. Farmer, S.A. Kauffman, N.H. Packard, A.S. Perelson and I.M. Stadnyk. 1989. Modeling adaptive biological systems. *Biosystems* **23**, 113–138.

Bagley, R.J. and J.D. Farmer. 1992. Spontaneous emergence of a metabolism. In *Artificial Life II*. Langton, C.G., C. Taylor, J.D. Farmer and S. Rasmussen (eds), pp.93–141. Redwood City: Addison-Wesley.

Bagley, R.J., J.D. Farmer and W. Fontana. 1992. Evolution of a metabolism. In *Artificial Life II*. Langton, C.G., C. Taylor, J.D. Farmer and S. Rasmussen (eds), pp.141–158. Redwood City: Addison-Wesley.

Banatre, J.-P., D. Le Metayer. 1986. A new computational model and its discipline of programming. Technical Report. INRIA Report 566.

Barendregt, H.G. 1981. *The Lambda Calculus: Its Syntax and Semantics.* Amsterdam: North Holland.

Berry, G. and G. Boudol. 1990. The Chemical Abstract Machine. In *17th ACM Annual Symposium on Principles of Programming Languages.* pp.81–94. New York: ACM Press.

Bjoerlist, M.C. and P. Hogeweg. 1991. Spiral wave structure in prebiotic evolution: hypercycles stable against parasites. *Physica D*, **48**, 17–28.

Buss, L. W. 1987. *The Evolution of Individuality.* Princeton: Princeton University Press.

Buss, L.W. 1994. Protocell life cycles. In *Early Life on Earth.* Bergstrom, S. (ed.). New York: Columbia University Press. In press.

Church, A. 1932. A set of postulates for the foundation of logic. *Annals of Math. (2)*, **33**, 346–366, and **34**, 839–864.

Church, A. 1941. *The Calculi of Lambda Conversion.* Princeton: Princeton University Press.

Church, A. and J.B. Rosser. 1936. Some properties of conversion. *Trans.Amer. Math.Soc.*, **39**, 472–482.

Curry, H.B. and R. Feys. 1958. *Combinatory Logic. Volume 1.* Amsterdam: North-Holland.

Curry, H.B., J.R. Hindley and J.P. Seldin. 1972. *Combinatory Logic. Volume 2.* Amsterdam: North-Holland.

Dyson, F. 1985. *Origins of Life.* Cambridge: Cambridge University Press.

Eigen, M. 1971. Self-organization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, **58**, 465–526.

Eigen, M. and P. Schuster. 1977. The hypercycle. A principle of natural self-organization. A: Emergence of the hypercycle. *Naturwissenschaften*, **64**, 541–565

Eigen, M. and P. Schuster. 1978a. The hypercycle. A principle of natural self-organization. B: The abstract hypercycle. *Naturwissenschaften*, **65**, 7–41

Eigen, M. and P. Schuster. 1978b. The hypercycle. A principle of natural self-organization. C: The realistic hypercycle. *Naturwissenschaften*, **65**, 341–369

Eigen, M. and P. Schuster. 1979. *The Hypercycle.* Berlin: Springer Verlag.

Eigen, M. and P. Schuster. 1982. Stages of emerging life – five principles of early organization. *J.Mol.Evol*, **19**, 47–61

Farmer, J.D., S.A. Kauffman and N.H. Packard. 1986. Autocatalytic replication of polymers. *Physica D*, **22**, 50–67.

Fisher, R.A. 1930. *The Genetical Theory of NaturalSelection.* Oxford: Clarendon Press.

Fontana, W. 1990. Algorithmic chemistry: a model for functional self-organization. Technical Report. SFI 90-011, Santa Fe Institute, Santa Fe, and Technical Report LA-UR-90-1959, Los Alamos National Laboratory, Los Alamos.

Fontana, W. 1991. Functional self-organization in complex systems. In *1990 Lectures in Complex Systems.* Nadel, L. and D. Stein (eds), pp.407–426. Redwood City: Addison-Wesley.

Fontana, W. 1992. Algorithmic chemistry. In *Artificial Life II*, Langton, C.G., C. Taylor, J.D. Farmer and S. Rasmussen (eds), pp.159–209. Redwood City: Addison-Wesley.

Fontana, W. and L.W. Buss. 1993. What would be conserved 'if the tape were run twice'. *Proceedings of the National Academy of Sciences, USA.* In press.

Forrest, S. 1990. Emergent computation: self-organizing, collective and cooperative phenomena in natural and artificial computing networks. *Physica D*, **42**, 1–11.

Fox, S. W. and K. Dose. 1977. *Molecular Evolution and the Origin of Life.* New York: Marcel Dekker.

Futrelle, R. P. and A.W. Miller. 1992. Simulation of metabolism and evolution using Quasichemistry. In *Biomedical Modeling and Simulation.* Eisenfeld, J., D.S. Levine and M. Witten (eds.), pp 273–280. Amsterdam: Elsevier.

Gould, S.J. and R.C. Lewontin. 1979. The spandrels of San Marco and the Panglossian paradigm: a critique of the adaptionist programme. *Proc.Roy.Soc., London.*, **B205**, 581–598.

Haldane, J.B.S. 1932. *The Causes of Evolution.* New York: Longmans, Green.

Haldane, J.B.S. 1954. The origins of life. *New Biology*, **16**, 12–27.

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

Hofbauer, J. and K. Sigmund. 1988. *The Theory of Evolution and Dynamical Systems.* Cambridge: Cambridge University Press.

Huet, G. and D.C. Oppen. 1980. Equations as rewrite rules. In *Formal Languages: Perspectives and Open Problems.* Book, R. (ed). New York: Academic Press.

Huet, G. 1992. *Constructive Computation Theory. Part I.* Notes de Cours. DEA Informatique, Mathématiques et Applications. (Available by anonymous ftp from ftp.inria.fr, /INRIA/formel/cct/CCT.dvi)

Jacob, F. 1982. *The Possible and the Actual.* New York: Pantheon Books.

Kauffman, S.A. 1971. Cellular homeostasis, epigenesis and replication in randomly aggregated macromolecular systems. *J.Cybernetics*, **1**, 71–96.

Kauffman, S.A. 1986. Autocatalytic sets of proteins. *J.Theor.Biol.*, **119**, 1–24.

von Kiedrowski, G. 1986. A self-replicating hexadeoxynucleotide. *Angew.Chem.* **98**, 932–934.

Knuth, D.E. and P.E. Bendix. 1970. Simple word problems in universal algebra. In *Computational Problems in Abstract Algebra.* Leech, J. (ed). pp.263–297. New York: Pergamon Press.

Lakoff, G. 1987. *Women, Fire, and Dangerous Things. What Categories Reveal about the Mind.* Chicago: University of Chicago Press.

Lane, D. 1993a. Artificial worlds and economics. Part I. *J.of Evolutionary Economics*, **3**, 89–107.

Lane, D. 1993b. Artificial worlds and economics. Part II. *J.of Evolutionary Economics.* In press.

Leifer, E.M. 1991. *Actors as Observers. A Theory of Skill in Social Relationships.* New York: Garland Publishing.

Lewontin, R.C. 1970. The units of selection. *Ann.Rev.Ecol.System.*, **1**, 1–18.

Lindgren, K. 1992. Evolutionary phenomena in simple dynamics. In *Artificial Life II*, Langton, C.G., C. Taylor, J.D. Farmer and S. Rasmussen (eds), pp.295–312. Redwood City: Addison-Wesley.

Lotka, A.J. 1925. *Elements of Physical Biology.* New York: Dover Reprint.

Luisi, P.L. 1993. Defining the transition to life: self-replicating bounded structures and chemical autopoiesis. In *Thinking About Biology.* Stein, W. and F.J. Varela (eds.), pp.3–23. Redwood City: Addison-Wesley.

Maturana, H. and F. Varela. 1973. *De Máquinas y Seres Vivos: Una teoría de la organización biológica.* Santiago de Chile: Editorial Universitaria. (Reprinted in english in Maturana and Varela, 1980.)

Maturana, H. and F. Varela. 1980. *Autopoiesis and Cognition: The Realization of the Living.* Boston: D.Reidel.

May, R. M. (ed) 1976. *Theoretical Ecology: Principles and Applications.* Oxford: Blackwell Scientific.

Maynard-Smith, J. 1982. *Evolution and the Theory of Games.* Cambridge: Cambridge University Press.

Maynard-Smith, J., R. Burian, S.A. Kauffman, P. Alberch, J. Campbell, B. Goodwin, R. Lande, D. Raup and L. Wolpert. 1985. Developmental constraints and evolution. *Quart.Rev.Biol.*, **60**, 265–287.

Miller, S.M. and L.E. Orgel. 1974. *The Origins of Life on the Earth.* Englewood Cliffs, NJ: Prentice-Hall.

Morowitz, H. J. 1992. *Beginnings of Cellular Life.* New Haven: Yale University Press.

Niesert, U., D. Harnasch and C. Bresch. 1981. Origin of life Between Scylla and Charybdis. *J.Mol.Evol.*, **17**, 348–353.

Newman, M.H.A. 1941. On theories with a combinatorial definition of "equivalence". *Annals of Math.*, **43**, 223-243.

Odifreddi, P. 1989. *Classical Recursion Theory.* Amsterdam: North Holland.

Oparin, A.I. 1924. *Proiskhozhdenie zhizni.* Moscow: Moskovskij Rabochij.

Padgett, J.F. and C.K. Ansell. 1993. Robust action in the rise of the Medici: 1400-1434. *Am.J.Soc.*, **98**, 1259–1319.

Penrose, R. 1989. *The Emperor's New Mind.* Oxford: Oxford University Press.

Revesz, G.E. 1988. *Lambda-Calculus, Combinators, and Functional Programming.* Cambridge: Cambridge University Press.

Rössler, O. 1971. Ein systemtheoretisches Modell zur Biogenese. *Zeitschrift für Naturforschung*, **26b**, 741–746.

Stadler, P.F., W. Fontana and J.H Miller. 1993. Random catalytic reaction networks. *Physica D*, **63**, 378–392.

Thürk, M. 1993. Ein Modell zur Selbstorganisation von Automatenalgorithmen zum Studium molekularer Evolution. PhD thesis. Friedrich-Schiller Universität Jena, Germany.

Tjiuikaua, T, P. Ballester and J. Rebek Jr. 1990. A self-replicating system. *J. Am. Chem. Soc.* **112**, 1249–1250.

Varela, F., H.R. Maturana, and R. Uribe. 1974. Autopoiesis: the organization of living systems, its chracterization and a model. *BioSystems*, **5**, 187–196.

Varela, F., A. Coutinho, B. Dupire and N.N Vaz. 1988. Cognitive Networks: Imune, Neural, and Otherwise. In *Theoretical Immunology, Part Two.* Perelson, A.S. (ed.), pp.359–375. Redwood City: Addison-Wesley.

Vrba, E. and S.J. Gould. 1986. Sorting is not selection. *Paleobiology*, **12**, 217–228.

DeVries, H. 1904. *Species and Varieties: Their Origin by Mutation.* Chicago: Open Court Publ. Co.

Winograd, T. and F. Flores. 1986. *Understanding Computers and Cognition.* Reading: Addison-Wesley.

Wright, S. 1931. Evolution in Mendelian populations. *Genetics*, **16**, 97–159.

10 Appendices

10.1 Remarks on axioms of lambda-calculus

The system defined in section 4.2 consists only of syntax. λ-calculus was, however, designed with an interpretation in mind. A few informal comments in this regard may be helpful in obtaining a better understanding of the syntactical system.

(1): Trivial.

(2): This axiom declares x to be a formal parameter of M, thus turning M into a function of x (x may or may not appear free in M). It “abstracts” a *procedure* (i.e. function) from M by generalizing the expression x in M from being a particular fixed symbol to being a “slot” (a bound variable) that may be replaced by anything.

(3): Application is the operation whose mechanics is defined in axioms (4) – (7). At the same time λ-calculus declares application as a *syntactical structure* of terms, thereby obtaining closure of Λ by definition: application is a term forming operation.

Axioms (4) - (7) define schemes of rearrangement that can be applied to any term in Λ. If a term contains a (sub)structure of the form specified on the left hand side, that (sub)structure can be replaced by the term on the right hand side. Taken together these axioms define “substitution” by induction over the structure of terms. Notice: the left hand side always has the form of an “application”, axiom (3), and that which is applied always has the form of an “abstraction” (that is: a function), axiom (2): application of a function pulls the trigger of syntactical transformations. An abstraction prefix is removed upon application. Roughly speaking, iterated application and iterated abstraction are inverses of each other.

(4): Trivial.

(5): x is designated a variable, but does not appear in the body E of the function, or is already bound by another abstraction. Q is substituted for x which is not available, and the $\lambda x.$ prefix is removed: a constant function.

(6): This axiom prepares the expression so as to make substitution occur for the first bound variable in case of multiple abstractions. The conditions ensure that no free variable in E or Q is captured in subsequent processing. Obviously one could choose the alternative form:

$$(\lambda x.\lambda y.E)Q \quad\longrightarrow\quad \lambda x.(\lambda y.E)Q,\ \text{if } x \neq y, \text{ and } (x \notin \phi(Q) \vee y \notin \phi(E)) \quad (6')$$

Both, (6) and (6'), are equivalent. To fulfill the conditions of (6) a renaming via (8) may be necessary.

(7) A kind of distribution law. If the function body consists of a term applied to another term, substitution of Q for x occurs in both. This axiom merely restructures the term for further processing by (4) – (7).

(8): Provides effective renaming of variables, when the term on the right hand side is submitted to (4) – (7) for processing.

10.2 Example of reduction to normal form

The following example illustrates the normal order reduction of

$$(((\lambda x.\lambda y.\lambda z.(y)x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)y.$$

The first column numbers the reduction step, the second column refers to the invoked rewrite axiom, and the third gives the resulting expression.

$$\begin{array}{lll}
0: & & (((\lambda x.\lambda y.\lambda z.(y)x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)y \\
1: & \xrightarrow{6} & ((\lambda \$_1.(\lambda x.\lambda z.(\$_1)x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)y \\
2: & \xrightarrow{6} & ((\lambda \$_1.\lambda z.(\lambda x.(\$_1)x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)y \\
3: & \xrightarrow{7} & ((\lambda \$_1.\lambda z.((\lambda x.\$_1)\lambda x.(x)y)(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)y \\
4: & \xrightarrow{5} & ((\lambda \$_1.\lambda z.(\$_1)(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)y \\
5: & \xrightarrow{6} & (\lambda z.(\lambda \$_1.(\$_1)(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)y \\
6: & \xrightarrow{7} & (\lambda z.((\lambda \$_1.\$_1)\lambda u.\lambda z.(u)z)(\lambda \$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)y \\
7: & \xrightarrow{5} & ((\lambda \$_1.\$_1)\lambda u.\lambda z.(u)z)(\lambda \$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z \\
8: & \xrightarrow{4} & (\lambda u.\lambda z.(u)z)(\lambda \$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z \\
9: & \xrightarrow{6} & \lambda z.(\lambda u.(u)z)(\lambda \$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z \\
10: & \xrightarrow{7} & \lambda z.((\lambda u.u)(\lambda \$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)(\lambda u.z)(\lambda \$_1.(\lambda x.x)\lambda x.(x)y) \\
 & & \lambda u.\lambda z.(u)z \\
11: & \xrightarrow{4} & \lambda z.((\lambda \$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z)(\lambda u.z)(\lambda \$_1.(\lambda x.x)\lambda x.(x)y)\lambda u. \\
 & & \lambda z.(u)z \\
12: & \xrightarrow{5} & \lambda z.((\lambda x.x)\lambda x.(x)y)(\lambda u.z)(\lambda x.x)\lambda x.(x)y \\
13: & \xrightarrow{4} & \lambda z.(\lambda x.(x)y)(\lambda u.z)\lambda x.(x)y \\
14: & \xrightarrow{7} & \lambda z.((\lambda x.x)(\lambda u.z)\lambda x.(x)y)(\lambda x.y)(\lambda u.z)\lambda x.(x)y \\
15: & \xrightarrow{4} & \lambda z.((\lambda u.z)\lambda x.(x)y)(\lambda x.y)(\lambda u.z)\lambda x.(x)y \\
16: & \xrightarrow{5} & \lambda z.(z)(\lambda x.y)z \\
17: & \xrightarrow{5} & \lambda z.(z)y
\end{array}$$

whose standardized normal form is: $\lambda x_1.(x_1)y$

1. According to normal order reduction we must first contract the leftmost redex. The leftmost redex is the substring $(\lambda x.\lambda y.\lambda z.(y)x)\lambda x.(x)y$. Hence axiom (6) is invoked, with $E \equiv \lambda z.(y)x$ and $Q \equiv \lambda x.(x)y$. However, $x \in \phi(E) \wedge y \in \phi(Q)$, and to fulfill the conditions we rename the y in E to, say, $\$_1$, for avoiding confusion with the y in Q. The renaming is executed via axiom (8). Since its effect is evident, we jump to the renamed expression.

2. Axiom (6) is a restructuring in preparation for a contraction. (In fact, it simply insures that the first bound variable will be substituted first.) After use of axiom (6) we must continue with the redex $(\lambda x.\lambda z.(\$_1)x)\lambda x.(x)y$ created inside the former by this restructuring; the so-called "trace" (for details see Revesz, 1988, pp.128–129). This requires again axiom (6), with $E \equiv (\$_1)x$ and $Q \equiv \lambda x.(x)y$. This time $x \in \phi(E) \wedge z \notin \phi(Q)$, hence the conditions for axiom (6) are fulfilled and no renaming is necessary.

3. We have invoked axiom (6), and hence must continue with its trace $(\lambda x.(\$_1)x)\lambda x.(x)y$. The appropriate reduction scheme is axiom (7), with $E_1 \equiv \$_1$ and $E_2 \equiv x$.

4. $(\lambda x.\$_1)\lambda x.(x)y$ is finally contracted with axiom (5), leaving $(\$_1)$.

5. A contraction has now been performed, and we continue with normal order, that is: with the leftmost redex in the expression. This redex is $(\lambda\$_1.\lambda z.(\$_1)(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z$, which invokes axiom (6) with $E \equiv (\$_1)(\lambda x.x)\lambda x.(x)y$ and $Q \equiv \lambda u.\lambda z.(u)z$. Since $\$_1 \in \phi(E) \wedge z \notin \phi(Q)$, no renaming is necessary.

6. The trace $(\lambda\$_1.(\$_1)(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z$ is processed by axiom (7) with $E_1 \equiv (\$_1)$ and $E_2 \equiv (\lambda x.x)\lambda x.(x)y$.

7. We continue with normal order. The leftmost redex is in fact the entire expression. The appropriate contraction scheme is axiom (5), since z does not occur free in $((\lambda\$_1.\$_1)\lambda u.\lambda z.(u)z)(\lambda\$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z$.

8. The leftmost redex is now $(\lambda\$_1.\$_1)\lambda u.\lambda z.(u)z$ which is contracted with axiom (1).

9. The leftmost redex is the current expression, which is prepared for further processing by axiom (6), with $E \equiv (u)z$ and $Q \equiv (\lambda\$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z$. No renaming is necessary.

10. The trace is $(\lambda u.(u)z)(\lambda\$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z$, and axiom (7) must be applied with $E_1 \equiv u$, $E_2 \equiv z$, and $Q \equiv (\lambda\$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z$.

11. The leftmost redex is $(\lambda u.u)(\lambda\$_1.(\lambda x.x)\lambda x.(x)y$. Axiom (4) yields its contractum.

12. The leftmost redex $(\lambda\$_1.(\lambda x.x)\lambda x.(x)y)\lambda u.\lambda z.(u)z$ is contracted with axiom (5).

13. The leftmost redex $(\lambda x.x)\lambda x.(x)y$ is contracted with axiom (4).

14. The leftmost redex $(\lambda x.(x)y)(\lambda u.z)\lambda x.(x)y$ is processed via axiom (7)

15. Contraction with axiom (4).

16. Contraction with axiom (5).

17. Finally, contraction with axiom (5) yields the normal $\lambda z.(z)y$, which is equivalent to $\lambda x_1.(x_1)y$.