A Category Theoretical Argument against the Possibility of Artificial Life: Robert Rosen's Central Proof Revisited

Dominique Chu
Computing Laboratory
University of Kent
Canterbury
United Kingdom
and
School of Computer Science
University of Birmingham
Birmingham B15 2TT
United Kingdom
D.F.Chu@kent.ac.uk

Weng Kin Ho
School of Computer Science
University of Birmingham
Birmingham B15 2TT
United Kingdom
W.K.Ho@cs.bham.ac.uk

Abstract One of Robert Rosen's main contributions to the scientific community is summarized in his book *Life itself*. There Rosen presents a theoretical framework to define living systems; given this definition, he goes on to show that living systems are not realizable in computational universes. Despite being well known and often cited, Rosen's central proof has so far not been evaluated by the scientific community. In this article we review the essence of Rosen's ideas leading up to his rejection of the possibility of real artificial life in silico. We also evaluate his arguments and point out that some of Rosen's central notions are ill defined. The conclusion of this article is that Rosen's central proof is wrong.

1 Introduction

One of the stated goals of artificial life is the implementation of living systems in silico and their simulation over an entire life cycle [5]. So far this undertaking has been at best partially successful.

Several researchers have built computational systems that display lifelike behaviors. Very famous examples are Ray's Tierra [21] and Adami's Avida [2]; those systems consist of adaptive computer programs that compete for limited resources, displaying interesting evolutionary effects reminiscent of living systems. Other examples of living systems in silico are Ono and Ikegami's self-assembling cells [18, 17], McMullin's SCL model [3, 15, 14], and Rasmussen et al.'s self-assembling lipid bilayers [20].

Probably the most famous of artificial life creatures is John von Neumann's self-reproducing machine [6, 7]. The center piece of this machine is the so-called *universal constructor*—a machine that, given enough supplies and a description of a machine D, can build D. Together with a control unit that regulates the sequence of events and a description of all parts of the von Neumann machine, the universal constructor is an essential element allowing self-reproduction of the entire machine (for an in depth discussion of the intricacies of von Neumann's design see [13]).

Those and many other systems show interesting lifelike behavior; nevertheless, few researchers would call any of them "living." It is widely acknowledged that the problem of research into artificial life forms is not only of a technical nature (how to design an artificial organisms) but at least equally of a conceptual nature: We do not know what life is; at least, we do not have a set of criteria that can be applied to a given system to decide whether or not it is alive.

There have been some attempts to formulate a general theory of life. One of the best-known approaches in this context goes back more than thirty years in time and has culminated in Maturana

and Varela's celebrated notion of *autopoiesis* [25, 12]. An autopoietic system is characterized by two main properties:

- It is separated from its environment by a boundary.
- It has an internal organization that is capable of dynamically sustaining itself (including its boundary); this internal sufficiency of the productional processes is often referred to as *closure* [4].

While autopoiesis has had considerable influence on various sciences (notably not including biology), there is no stringent theory of autopoiesis that formulates its contents in an unambiguous way (see Nomura [16] for an attempt).

A more mathematical approach to the topic, also emphasizing closure properties, is due to Robert Rosen and is comprehensively laid out in his book *Life itself* [23]; also see [24, 22, 7, 8, 26, 9]. Rosen's work defines a living system as closed with respect to "efficient causation." This is often understood to be roughly equivalent to the closure property of autopoietic systems [11]. However, the two concepts are not identical. Maturana and Varela themselves illustrated the concept of autopoiesis by means of a computational model [25]; this model has since been re-implemented by McMullin and Varela [15, 14, 3]. Rosen's central message, on the other hand, is precisely that closed systems are fundamentally different from computing machines. Rosen takes one further step by claiming that living systems cannot even be implemented in computational systems. If he were right, this would mean that attempts to construct life in silico are futile. Altogether, Rosen's central conclusion is that life in silico is impossible.

A detailed presentation of the argument supporting his conclusion is laid out in Rosen's book *Life itself*; a central part of this book is a semi-formal proof for this assertion. *Life itself* was published in 1991 and has since attracted quite considerable interest, reflected in numerous citings of it. This interest is not surprising, given that if the proof (and the conclusion) were correct, then this would have deep implications for the field of artificial life and even our view and understanding of life. Unfortunately, a clean assessment of the argument is hampered by the inaccessibility of Rosen's writing. Its free mixing of mathematical allusions, formal arguments, and comments on the history and philosophy of science, in addition to poor editing, makes it very hard for the reader to distill the essential argument from the text, let alone critically assess it. Rosen's idiosyncratic style is not helpful in this effort.

In this article we aim at two objectives. Firstly, we wish to clarify a number of concepts that are essential for Rosen's proof but not clearly presented by Rosen himself. In order to achieve this we will (in Section 2) review the notions of synthetic and analytic models. Those concepts are central to Rosen's idea of machine versus living system, and thus essential for his central proof. An important part of this section will also be to clarify the relations between those two models and certain concepts from category theory. This relation is only hinted at in Rosen's original work. Having clarified those basic concepts of synthetic and analytic models, we will then go on to define "mechanisms" (and "machines") in Section 3. The notion of mechanism is of particular importance in that it is coextensive with the set of systems that can be simulated in computers. In Section 4 we will recap Rosen's central proof stating that living systems are not mechanisms, hence not simulable.

The second main goal of this article is to give a critical assessment of Rosen's central proof. We will do this in Section 5, where we also show that Rosen's own concept of machine is flawed. We will provide a brief discussion and conclusion in Section 6.

2 Models

In this section we will introduce Rosen's notion of synthetic and analytic model.

Rosen focused his discussion entirely on models; he largely avoided writing about systems. The reason for this is that as observers we do not have direct access to real systems in the physical world

(the *Welt an sich*). Knowledge about the real world is thus necessarily incomplete; at the very least, whatever we know about real systems, we can never be sure what the status of this knowledge actually is.

This difficulty disappears when we restrict the discussion to formal systems instead; in contrast to real systems, formal systems are accessible. Throughout the remainder of this article, the word "system" will always refer to formal systems unless explicitly stated otherwise. For all practical purposes, thus, the notion of system is interchangeable with the notion of model, only that we tend to use the word "system" to denote given formal systems and their very accurate models.

Rosen's argument fundamentally rests on the distinction between two types of models, namely *synthetic* and *analytic* models [23, 22]. The main purpose of this section is to introduce them and to clarify their relation to certain concepts from category theory. Since we do not assume that the reader is familiar with this branch of mathematics, we will give a brief introduction to some concepts of category theory. An exhaustive discussion of this topic would of course go beyond the scope of this article; the reader who wishes to explore category theory in any depth is referred to textbooks [19, 10].

2.1 Categories

We start by defining the notion of category.

A *category* is a construct that consists of the following:

- Objects (sets, or anything else; in our case: models).
- Morphisms between objects; not every object needs to be the domain or codomain of a morphism.
- For every object X, an identity morphism id_X.
- For each pair of morphisms $f : A \to B$, $g : B \to C$ a composite morphism $fg : A \to C$.
- Furthermore, we require associativity $[(fg)h = f(gh)]$ and that morphisms can be combined with the identity map (given a morphism $f : A \to B$, we have $f\,id_A = f$ and $id_B f = f$).

In the simplest case, the objects of a category are sets and the morphisms are mappings between the sets. In the more general case the objects might be anything, including categories themselves. Categories are often graphically represented as points (objects) of which some are connected by directed graphs (the morphisms), usually indicated as arrows. The strength of category theory is that it allows one to formulate relations between the objects and also relations between categories.

A pervasive concept in category theory is the idea of duality. Two concepts in category theory are *dual* if one is like the other but with all arrows reversed. For example, if A is a category, then the opposite category A^{op} is A with all arrows reversed. A and A^{op} are then dual to each other. Throughout this article the concept of duality will not be explored in any depth, but it will be used to indicate relations between definitions and categories; specifically, the concepts of direct sum and direct product are dual; the reader might compare the definitions in Figures 1 and 3.

The reason the direct product and sum need to be introduced here is that Rosen essentially identifies them with analytic and synthetic models respectively. The concepts of analytic and synthetic model in turn are essential to the concept of *mechanism*—the class of systems that can be simulated in computers.

2.2 Analytic Models

While the concepts of analytic and synthetic model are crucial to the entire argument, Rosen only hinted at a mathematical definition. The purpose of this section is to make the notion of analytic model precise.

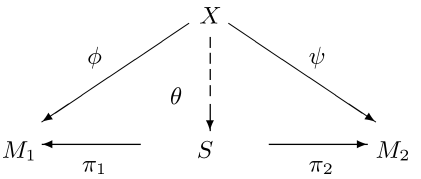

Figure 1. Direct product: If S, M_1, M_2 are objects of the category C, then S is the direct product if we can find an object X and morphisms $\pi_1 : S \to M_1$, $\pi_2 S \to M_1$, $\phi : X \to M_2$, and $\psi : X \to M_2$ such that there is a *unique* morphism $\theta : X \to S$. The direct product is a generalization of the familiar concept of the Cartesian product of two sets to arbitrary categories.

Before doing so, however, we will briefly discuss the informal idea behind analytic models. Rosen's notion of model might to some appear somewhat counter-intuitive at first. A typical model in science formulates a dynamic relationship between entities, as for example a differential equation. Rosen's idea of model is fully compatible with this conventional view, yet somewhat different in emphasis. Instead of focusing on the dynamics of components of a system, a model in Rosen's sense is a certain way to measure properties of the system. Given a system S, an analytic model of S is a set of observables that describe the system. Clearly, the quality of the description depends on the accuracy with which the observables are measured. An observable that can have say, 100 different values, is better, or at least of higher resolution, than an observable that can have only two values. Increasing the resolution of an observable thus leads to a *refined* model.

Besides an increase of resolution of an observable, there is a second type of refinement: Not only might it be that the observables used are inaccurate, measuring the system only crudely; it might also be that the set of observables used is not sufficient to give a full description of the system. In this situation adding a new observable will also refine the model. One might for example describe a car in terms of its speed and the number of passengers it takes; this description would, in Rosen's sense, be a simple analytic model. Another model of a car would describe it in terms of its milage and how much fuel it carries. Putting those two models together, thus describing the car in terms of speed, number of passengers, milage, and fuel content, will give a more accurate picture of the state of the car than either of the models and thus refine both of them.

For every system S there will usually be many different models that are in a partial order relationship with each other, generated by the refinement relationships between them. Note that, while models in Rosen's sense do not explicitly take into account the dynamics of the system, they nevertheless are complete descriptions of the system. This reflects the assumption that a complete description of a system at a specific time already determines all future behavior (although possibly only in a statistical sense). In essence, an analytic model is a set of observables with a given resolution.

Formally, an observable is a mapping f from a system S (the system to be modeled) to a set (usually $\mathbb{R}^n$). The observable f induces an equivalence relation on S: Two elements $s_1, s_2 \in S$ are ***equivalent*** ($s_1 \sim s_2$) if $f(s_1) = f(s_2)$. The size of the equivalence classes thus indicates how well f discriminates between states of the system; the smaller the equivalence classes, the better is the corresponding observable. An ***analytic model*** M of S is the partition of S into equivalence classes generated by some f.

One and the same system S will typically have many different analytic models corresponding to all possible observables f; those models will be related to each other and can be compared by the way they partition S into equivalence classes. The relevant concept here is that of refinement: Given a model M_1, the model M_2 is a ***refinement*** of M_1 if all equivalence relations induced on S by M_2 are subsets of the equivalence relations induced by M_1 on S. Intuitively, this means that M_2 can distinguish between at least as many states of S as M_1. Given two models M_1, M_2 of S, those models need not be in a refinement relation to each other, but may be *unlinked*.[1] This will be the case if the observables measure unrelated aspects of the system. If there is no specific refinement relation between M_1

1 It is an important technicality that observables and hence models may be partially linked. This is not of fundamental importance, though, and will be ignored in the present discussion; the reader interested in this aspect is referred to Rosen's discussion in [22].

and M_2, it is always possible to construct a model M_3 that is a refinement of both M_1 and M_2 by taking the Cartesian product of the two models (strictly, the Cartesian product of the respective equivalence classes on S): If $f_1 : S \to \mathbb{R}^{k_1}$ and $f_2 : S \to \mathbb{R}^{k_2}$ are the observables belonging to the models M_1, M_2 respectively, then the refined model $M_3 = M_1 \otimes M_2$ (the mutual refinement of M_1, M_2) is induced by a new observable $f_3 = (f_1, f_2) : S \to \mathbb{R}^{k_1} \times \mathbb{R}^{k_2}$. The set of equivalence classes on S induced by f_3, on the other hand, is the Cartesian product of the equivalence classes induced by f_1 and f_2 respectively.

Example. The system is the state space of a resting point mass in a two-dimensional discrete space. Possible models of this system are:

M_{x_1} $f_{x_1} : S \to \{0, 1\}$

M_{x_2} $f_{x_2} : S \to \mathbb{N}$

M_y $f_y : S \to \mathbb{N}$

The observable f_{x_1} distinguishes only between two different positions of the particles in the space. The corresponding model M_{x_1} is thus refined by the model M_{x_2}, which distinguishes between countably many positions. There is no refinement relation between M_y and either M_{x_1} or M_{x_2}. The mutual refinement of those models can easily be constructed:

M_{xy} $f_{xy} = (f_{x_1}, f_y) : S \to \mathbb{N} \times \mathbb{N}$

Formally, a set of analytic models of S and the refinement relations between them are a category $\mathcal{A}(S)$: The objects of the category are the models, and the morphisms are the refinement relations between them (for the detailed definition see Figure 2). Formally, this category looks like the category of partially ordered sets (*posets*; see Figure 2). This structure will turn out to be of high relevance for Rosen's central argument. Note that a system S is typically associated with many categories of analytic models; for every system S, $\mathbb{M}^a(S)$ is the set of all categories of analytic models of S. One element of $\mathbb{M}^a(S)$ will be the category consisting of the most refined model as the only object. Others will contain the most refined one and less refined ones plus the refinement relations between them; yet other categories will lack the most refined model of S. There are no restrictions on the possible categories, but note that we always assume that at the crudest level of modeling all observables are totally unlinked (that is, the bottom objects in the category are induced by unlinked observables).

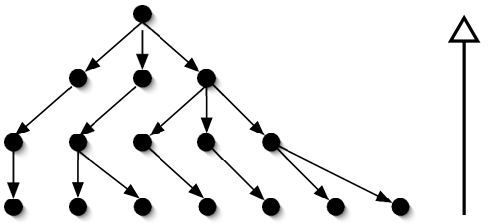

Figure 2. The category of analytic models: The objects of the category are models. The morphisms are refinement relations. There is a morphism from object M_1 to M_2 if M_1 is a refinement of M_2. Note that every model is its own refinement; thus for every object the identity (*id*) map exists. In this specific example, the top-level model is the direct product of any pair of second-level objects. To see this, identify S and X with the top-level object. The projections π_2 and π_1 are identical with the morphisms ϕ and ψ respectively. Then there is a unique morphism from S to S, as required by the definition. This unique morphism is the identity on S.

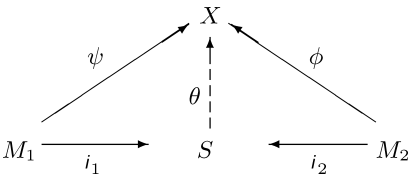

Figure 3. Coproduct: The coproduct (direct sum) is the dual to the product. It is defined by reversing all arrows in Figure 1. If S, M_1, M_2 are objects of the category C, then S is the coproduct of M_1 and M_2 if we can find an object X and morphisms $i_1 : M_1 \to S$, $i_2 : M_2 \to S$, $\psi : M_1 \to X$, and $\phi : M_2 \to X$ such that there is a *unique* morphism $\theta : S \to X$.

Rosen points out that the idea of analytic model is tightly connected to the concept of direct product. (For the definition of a direct product see Figure 1.) This is true, but only partially so. The mutual refinement of two models that are based on unlinked observables is formally the direct product in the category $\mathcal{A}(S)$. In the above example the model $M_3 = M_1 \otimes M_2$ is defined as the Cartesian product of the models M_1 and M_2; the reader can easily convince himself that in the corresponding category of analytic models consisting of the objects $\{M_1, M_2, M_3\}$ and the corresponding inclusion relations between them, M_3 is indeed the direct product of M_1 and M_2 according to the definition in Figure 1.

However, not all models in $\mathcal{A}(S)$ will in general be direct products of other models in $\mathcal{A}(S)$. Trivially, the bottom objects—the crudest models—will not. This can easily be seen to be the case by comparing Figure 2 with the definition of the direct product in Figure 1. More important is another case: The refinement of the resolution of an observable (thus not the mutual refinement of two observables) will lead to linear branches in the category; there will be no direct products involved.

2.3 Synthetic Models

The second kind of models are synthetic models. In a sense, synthetic models are the opposite of analytic models. The concept of analytic model captures the idea that a modeler analyzes a specific given system into parts (or, more precisely, into observables). This procedure is often referred to as a top-down approach. Synthetic models in Rosen's sense, on the other hand, correspond to bottom-up models. The intuitive idea behind synthetic models is as follows: Given two systems (interacting or not), one can (formally) unite both into a new, larger system containing the two components as subsystems. This can of course be generalized to any number of systems.

Most artificial life researchers will be very familiar with synthetic models. It is a common approach in the field to start out with a set of well-defined agents (usually simulated agents) and let them interact. The aim of such models is to understand the aggregate behavior of the agents.

Let us now give a more precise definition of a synthetic model: A ***synthetic model*** M is a poset of analytic models plus an order relation called *inclusion*. The following conditions need to be fulfilled:

1. The poset consists of at least three elements.
2. The poset has exactly one top element, that is, one element that is not itself included.
3. All but the lowest-level models in the poset are themselves synthetic models.

It follows from the definition that every element of the poset (except the top element) is included in exactly one other element, and every element (except for the bottom elements) includes at least two other elements. Graphically, a synthetic model thus has the form of a tree where every element is a branching point.

We will now describe the procedure to construct a synthetic model M_3 given the models M_1 and M_2. In order to do this, two cases need to be distinguished. Firstly, both M_1 and M_2 are not already synthetic models, but rather analytic models. The synthetic model M_3 can then be constructed by

taking their mutual refinement $M_1 \otimes M_2$; in this case the synthetic model is a partially ordered set with the elements $\{M_1, M_2, M_1 \otimes M_2\}$.

In the case that M_1 or M_2 or both are already synthetic models, M_3 can also be constructed. The unique top element of M_3 is then the mutual refinement of the top elements of M_1 and M_2; M_3 thus consists of the disjoint posets M_1 and M_2 joined together by a new top element. The construction of a new synthetic model is illustrated in Figure 4. All synthetic models are built up from atomic systems in this way; the peculiar structure of synthetic models is a consequence of the way they are constructed from the bottom up.

Like analytic models, synthetic models also form a category, $\mathcal{Y}(S)$ (see Figure 5). The objects of the category are synthetic models, and thus posets; those posets themselves form a new, higher-level poset. The morphisms between them are defined as order relations as follows: M_1 is *smaller* than M_2 if M_2 is a subtree of M (see Figure 5). The category $\mathcal{Y}(S)$ thus records the construction history of the synthetic model. All categories $\mathcal{Y}(S)$ have a very specific structure. The characteristic properties of $\mathcal{Y}(S)$ are:

- $\mathcal{Y}(S)$ has exactly one top object.
- All but the bottom objects include at least two synthetic models (the bottom objects include none).

Henceforth we will say that a category $\mathcal{D}$ ***looks like*** a synthetic model if it has this structure. Furthermore, we will denote by $\mathbb{M}^s(S)$ the set of all categories of synthetic models of S.

Given a category of synthetic models $\mathcal{Y}(S)$, it is immediate from the definition that a synthetic model M is the direct sum of M_1 and M_2 if M_1 and M_2 are one level below M and there are morphisms $\phi : M_1 \to M$ and $\psi : M_2 \to M$ (see Figure 5). Formally, thus, a model M is a synthetic model of M_1 and M_2 if it is of the form $M = M_1 \oplus M_2$; the models M_1 and M_2 may themselves be synthetic models. Note that the generalization of this discussion to any number of systems is straightforward. An example will illustrate what it means to take the direct sum of two models and, more important, what it does not mean.

EXAMPLE: Given two systems corresponding to point masses that live in a one-dimensional continuous space. The state space of each of them is two-dimensional, the dimensions corresponding to the position and the momentum respectively. The state space of the compound system is four-dimensional, describing the momenta and the positions of both particles. Establishing a new description uniting the two separate point masses into one model is clearly synthetic modeling, yet formally, the state space of the larger system is the Cartesian product of the state spaces of the individual particles and not the direct sum. (The direct sum would be the disjoint union of the state spaces of each of the particles and would represent a situation where either of the two particles is present in the system but not both of them.) It thus seems wrong to associate synthetic models with the direct sum.

This apparent error rests on a misunderstanding: Taking the direct sum of systems is not the same as taking the direct sum of their respective state spaces. Formally, the direct sum of two posets is just the disjoint sum of the posets, plus an arbitrary bottom element that connects the top

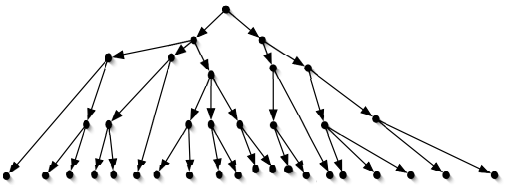

Figure 4. Synthetic models: A synthetic model is a poset of analytic models with a specific structure (see main text): All points need to be branching points, because the models at higher levels are the sums of at least two lower-level models.

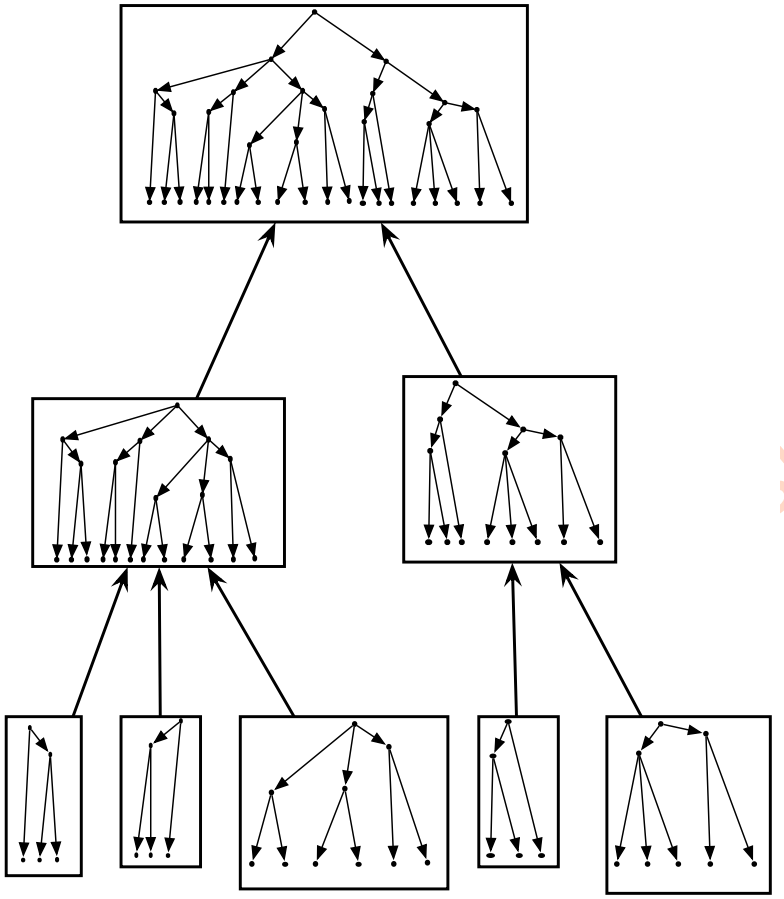

Figure 5. The category of synthetic models, $\mathcal{Y}(S)$: The objects of this category are synthetic models; those posets themselves form a new, higher-level poset. The morphisms between the objects are order relations. There is a morphism from object M_1 to object M_2 if M_2 contains M_1 as a subtree. Note that every object has a morphism onto itself. In the specific example in this figure it is thus easy to see that the top-level model is the direct sum of the second-level objects.

elements of the summands. This formally introduced bottom element in the new category has a very precise interpretation in this case: It is the direct product of the models M_1 and M_2 and thus the mutual refinement of M_1 and M_2. In summary: Taking the direct sum of two models induces a new system that is the direct product of the original models; the equivalence class induced by $M_1 \oplus M_2$ on S is the direct product of the equivalence classes induced by M_1 and M_2 respectively.

Finally, note that synthetic models are completely determined by their components (they are nothing but the sum of their parts). Also, the process of taking the direct sum of analytic models is a purely syntactic (or algorithmic) process. Given disjoint analytic models of systems, one can always construct a synthetic model by taking the mutual refinement between those models. This does not require new measurements of the system, but is an entirely formal procedure.

2.4 Relation between Analytic and Synthetic Models

Rosen defines mechanisms as the class of systems of which all analytic and synthetic models are equivalent; according to Rosen, this is the case when direct product and direct sum coincide. Again, Rosen does not actually spell out in detail what precisely it means for a synthetic model to be equivalent to a analytic model, nor does he say precisely what it means that direct sum and direct

product coincide. The purpose of this subsection is to clarify in which sense analytic and synthetic models might be equivalent.

Roughly, equivalence of synthetic and analytic models should mean that every synthetic model of a system S can be seen seen as an analytic model and every analytic model can be seen as a synthetic model. From the definition of the two types of models, it is clear that the phrase "can be seen as" cannot possibly mean "is identical to," because synthetic and analytic models are generally not the same type of things (synthetic models are posets of analytic models). The strategy we use here to define equivalence is to look at the structures of $\mathbb{M}^s(S)$ and $\mathbb{M}^a(S)$ (defined in Sections 2.3 and 2.2) and to establish a mapping between them. More specifically, we will define a subset $\mathbb{C}$ of $\mathbb{M}^a(S)$ and specify a mapping from $\mathbb{C}$ to $\mathbb{M}^s(S)$. Equivalence of synthetic and analytic models will then be defined in terms of the properties of this mapping.

Before doing so, however, let us look at the simpler case of relating synthetic models to analytic models. Rosen points out that the first part of the equivalence is always true, that is, synthetic models are always analytic models. Again, this is not to be understood in the sense of identity, but as follows: For every category of synthetic models $\mathcal{Y}(S) \in \mathbb{M}^s(S)$ the dual $\mathcal{Y}^{\mathrm{op}}(S)$ looks like a category of analytic models of $\mathcal{A}(S) \in \mathbb{M}^a(S)$ in the following sense: Consider this algorithm (see Figure 5):

1. Replace each object in the category of synthetic models with its top node.
2. Invert the arrows between the objects.

The reader can easily convince herself that this algorithm maps every element $\mathcal{Y}(S) \in \mathbb{M}^s(S)$ to a unique element $\mathcal{A}(S) \in \mathbb{M}^a(S)$ of the set of categories of analytic models of S. Note that the application of this algorithm is a purely syntactic procedure and does not require any additional information about the system, such as for example new measurements of the system. Thus every synthetic model can be transformed into an analytic model, or, as Rosen formulates it, every synthetic model is an analytic model.

The converse of this statement, which would complete the equivalence between synthetic and analytic models, is not true in general: Not every analytic model is a synthetic model. In order to see this, we will first have to determine what it means for a analytic model to be a synthetic model and more general for all analytic models of S to be synthetic models. Once we have clarified this, we will have to discuss under which circumstances all analytic models of S are actually synthetic models. Neither aspect is elaborated by Rosen himself.

Again, our informal understanding of an analytic model being a synthetic model is that for every $\mathcal{A}(S) \in \mathbb{M}^a(S)$ the dual $\mathcal{A}^{\mathrm{op}}(S)$ looks like a $\mathcal{Y}(S) \in \mathbb{M}^s(S)$. Two conditions need to be fulfilled in order for this to be the case: $\mathcal{A}(S)$ needs to have a unique top node, and this top node needs to join disjoint trees; this is so because by construction categories of synthetic models always have a unique top object.

Firstly, we assume that subtrees in the category of analytic models are always disjoint. This assumption only means that the observables at every level are mutually totally unlinked. A change of observables can always achieve this situation; since we are only interested in properties of systems, and not in properties of specific encodings of systems into observables, this restriction is acceptable and does not pose any restriction on the generality of systems about which we make statements.

Secondly, if a category $\mathcal{A}(S)$ has several top nodes, then we can always reduce the number of top nodes by taking the mutual refinement of the existing top nodes. Note that this is a purely syntactic procedure (in the context of real systems, we would not have to perform new measurements, only process the results of measurements we already have made).

There is another possible complication. The category of analytic models of S might contain non-branching refinement relations between models, that is, branches of the tree that look like $M_{k_0} \to M_{k_0+1} \to \cdots \to M_{k_0+n}$. In this case the model M_{k0} is associated with the same observable as the models M_{k_0+j} $(j = 1, 2, \ldots, n)$, but its resolution is higher than any of theirs. If the category of

analytic models contains such subtrees, then, formally $\mathcal{A}^{\mathrm{op}}(S)$ will not look like the dual of a category of synthetic models of S. However, from a practical point of view this is no problem. All the information about the system contained in the models M_{k_0+j} is already contained in M_{k_0}, and the models themselves can be recovered from M_{k_0} at any time. We can therefore remove all but the most refined model in the tree, without losing information about the system. Doing this for all linear branches, we remove this complication. Again, this is a purely algorithmic procedure; if there are linear branches in the model, then this says nothing about the system, but rather records the specific refinement history.

A situation where the current picture breaks down arises in the case where there is no finite set of observables that will lead to a complete description of S. In this case, there will be no maximally refined model. Thus finality of the system is a necessary condition for analytic models to be synthetic models.

Let now $\mathbb{C}$ be the reduced set of categories $\mathcal{A}(S)$ of analytic models. We obtain $\mathbb{C}$ from $\mathbb{M}^a(S)$ as follows:

- For all elements $\mathcal{A}(S) \in \mathbb{M}^a(S)$ do:
  1. If $\mathcal{A}(S)$ contains multiple top nodes, add new top nodes by taking the mutual refinement of existing top nodes.
  2. Replace all linear branches in $\mathcal{A}(S)$ with the most refined model in the linear branch.
- If the resulting category is not already in $\mathbb{C}$, add it.

Henceforth, we will say that all analytic and synthetic models of a system S are ***equivalent*** if for all categories $\mathcal{A}(S) \in \mathbb{C}$ the dual of $\mathcal{A}(S)$ looks like a category in $\mathbb{M}^s(S)$.

2.4.1 Analytic-Synthetic Means *S* Is Defined on a State Space

For the following we say that a system S is state-space-based if there is a set of observables O and a time t_0 such that the values of all $o \in O$ at t_0 completely determine the behavior of S for all $t > t_0$ (though possibly only stochastically).

It is straightforward to show that for all state-space-based systems S, analytic and synthetic models will always be equivalent: Having excluded sets of partially linked observables, all elements of $\mathbb{C}$ are, by construction, guaranteed to look like synthetic models; see the discussion above. The finite dimensionality of the state space guarantees that the construction of $\mathbb{C}$ can be done in finite time.

Similarly, the converse can be shown. If synthetic and analytic models are equivalent, then S has a unique most refined model in terms of a finite number of observables. Thus there are a finite number of unlinked observables that specify the system completely; this in turn means precisely that the systems is defined on a finite state space.

In summary, we conclude that:

- All synthetic models are analytic models.
- The converse does not hold in general, but holds for all systems that are defined on a finite state space.

Furthermore, whenever a system is defined on a state space, its synthetic models are equivalent to its analytic models. More important is the contrapositive statement: Whenever analytic models are not equivalent to synthetic models, the corresponding system S is not defined on a state space.

In *Life itself*, Rosen makes the connection between the equivalence of synthetic and analytic models and the equivalence of direct sum and direct product. The equivalences are indeed related to some extent—not, however, as closely as it might seem from Rosen's exposition. Since the concepts of direct product and direct sum are dual to each other, the following is always true: If a, b, c

are objects of a category $\mathcal{D}$ and if $a = b \otimes c$ in $\mathcal{D}$, then in the dual category $\mathcal{D}^{op}$ it will be true that $a = b \oplus c$. This, however, does not translate directly into a theorem about the equivalence of analytic and synthetic models, because, as the preceding discussion has made clear, the relation between synthetic and analytic models is only partially a relation of duality: While it is true that what is the direct product in a category $\mathcal{A}(S)$ will be a direct sum in $\mathcal{A}^{op}(S)$, this dual category will not always be a category of synthetic models of S.

3 Mechanisms and Machines

In this section we will now finally define the notion of mechanism. Mechanisms are an important class of systems, because Rosen defines them as the class of systems that are simulable; the question whether life in silico is possible is thus essentially the question whether or not at least some living systems are mechanisms. Rosen's central result in *Life itself* is that whatever has the organization of living systems (closure) cannot be a mechanism.

Rosen bases his definition of mechanisms on the concept of simulability. In order to avoid burdening the present discussion with yet another new concept, we will give an alternative, yet essentially equivalent definition of mechanism: A system is a ***mechanism*** if all its analytic models are equivalent to synthetic models. Thus, all mechanisms can be exhaustively defined as state-based systems. Mechanisms are always simulable in Rosen's sense [though they might not be simulable (or computable) in the colloquial sense of the word].

One class of mechanisms that are of specific interest for Rosen are what he calls *machines*; machines in Rosen's sense are essentially Turing machines. Traditionally Turing machines are thought of as consisting of a reading head—that is, a finite-state automaton that can read from and write to a tape. Rosen stipulates that "Turing machines" are machines *sensu* Rosen and thus mechanisms. Therefore they can be completely described in terms of states, $\{x_1, x_2, \ldots, x_{k-1}, x_k, \ldots, x_n\}$.

The reader might find the claim that a Turing machine can be exhaustively described in terms of states alone somewhat surprising. The best way to convince oneself that Turing machines are indeed mechanisms is to consider implementations of Turing machines in cellular automata (such as in the Game of Life [1]). All configurations in cellular automata are quite obviously state-space-based. In the case of the Game of Life, this description will be entirely in terms of a collection of binary states. If S is the $n \times m$ dimensional array of automaton cells, and $S' \subseteq S$ a k-cell subset of S, then the Turing machine (or in fact any other configuration) implemented in this CA can be completely specified by the following analytic model:

$$M_T \quad \mathtt{f} = (\mathtt{f}_1, \ldots, \mathtt{f}_k) : S' \rightarrow \{0,1\}^k$$

Rosen points out that models of machines in terms of states are not particularly instructive, and introduces an alternative representation, so-called *relational models* (see Figure 6). Relational models are a way to represent systems by showing how higher-level parts of the system interact. In the case

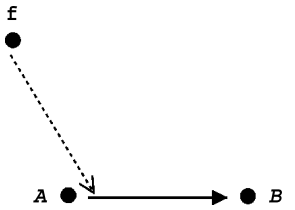

Figure 6. Rosen's (relational) model of a Turing machine. The vertex $\mathtt{f}$ denotes the hardware that induces a software flow from input (*A*) to output (*B*). Based on Figure 9B.3 in [23]. Note that the component $\mathtt{f}$ in this figure is different from the observable label *f* used in Section 2.2.

of a Turing machine a state description can be partitioned into states that correspond to hardware (say the first k states) and software, comprising input, output, and other software states:

$$\underbrace{x_1, x_2, \ldots, x_{k-1}, x_k}_{\text{hardware}}, \underbrace{x_{k-1}, \ldots, x_n}_{\text{software}}$$

In the relational model the hardware states are lumped together (component f in Figure 6). Similarly, input and output states are combined together into the components A and B respectively. The basic idea reflected in the relational model is that the software flow from input to output is implemented by the hardware components[2] f. Rosen points out that this flow of data can formally be seen as a mapping $f : A \to B$. In contrast with pure mathematics, where the mapping f is simply assumed, in real systems f needs to be realized by something in the world. In the relational model of the Turing machine this is reflected by the separate representation of f as a part of the machine (the hardware). In Figure 6, the solid arrow indicates that A is mapped into B, whereas the dashed arrow from f to the solid arrow from A to B indicates that this mapping is implemented by the component f; for reasons that will become clear below, we will henceforth express the fact that B is the codomain of a mapping $f : A \to B$ by saying that B is *maintained* by the mapping f. We will also say that the component f *implements* the mapping $f : A \to B$.

Relational models such as the one in Figure 6 play an important role in Rosen's work. Unfortunately, it is not clarified by Rosen how in general a complete (non-relational) description of a specific system S can be transformed into a relational one; in other words, it is unclear how, given a system, the mappings and their implementations can be recovered from, say, a state-space-based description of the system. The precise meanings of abstract diagrams such as the one in Figure 6 must thus remain somewhat unclear. However, in order to be able to present and evaluate Rosen's central argument, we will pretend for the moment that this is not a problem and that we have a full operational understanding of relational models. In Section 5 we will reconsider this point.

4 Rosen's Central Argument about Living Systems and Machines

The central result of *Life itself* is the proof that living systems are not mechanisms; any description of these systems in terms of states would thus be incomplete. Rosen introduces his notion of a living system in terms of general properties of relational models (closure). The central argument about the impossibility of artificial life in silico will then essentially consist in showing that the type of relational model that defines a minimal organism is inconsistent with the assumption that the corresponding system is fully defined by a state description. This also implies that the system cannot be implemented in a finite-state machine. This insight then substantiates Rosen's central result in *Life itself*, namely, that there are systems that are not mechanisms, and that in fact nearly all systems fail to be mechanisms. In this section we will recap Rosen's central argument.

Rosen's strategy to prove that organisms are not mechanisms is as follows: Firstly he defines a universal property of organisms. As in the case of autopoiesis, this property is closure (as defined below). He then shows that systems with this closure property cannot be modeled in a state-space system. Thus, organisms are not machines.

We will start with a discussion of closure. A system is ***closed*** in Rosen's sense if every component in the system except for a set of input components is maintained and every mapping in the system is implemented by a component within the system. (We see here the similarity to autopoiesis.) An example of a closed system is pictured in Figure 7, whereas Figure 6 is an example of a system that is not closed: The component f in Figure 6 implements the mapping $A \to B$, but it

2 The reader interested in a detailed description of machines in terms of mechanisms is referred to the original literature [23, Chaps. 8,9].

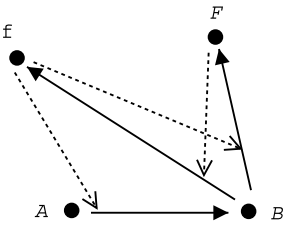

Figure 7. A minimal model of a system with closure. The dashed arrows indicate the implementation of a mapping, whereas the solid arrows indicate maintenance of a component. The component *A* represents input to the system. Note that this figure does not depict a category (as for example in Figure 5), but rather a possible object of the category of models.

is not itself maintained from within the system. Rosen then goes on to identify closure as a necessary condition for a system to be living.

The intuitive idea behind this is that systems that are not closed in this sense have some distinctively unbiological properties. For example, the system in Figure 6 must rely on external maintenance, because the mapping $\mathtt{f}$ is not maintained from within the system. If $\mathtt{f}$ is damaged it will go unrepaired, ultimately ceasing to function. This in turn leads to a disintegration of the system. In contrast, components in organisms are usually maintained by some other components within the organism. This should hold true for every component of the system, including the repair components themselves. The reader interested in a more exhaustive discussion of closure and its relevance for biological systems is referred to the relevant literature—in particular *Life itself*, but also [24, 4, 11].

We will now show how the system in Figure 6 can be enhanced (and closed); this discussion closely follows Rosen's own exposition in *Life itself*. First of all, in order to have $\mathtt{f}$ maintained, a new map needs to be added to the system, $F : B \to \mathtt{f}$. This enhancement leads to $\mathtt{f}$ being maintained, however, at the expense of a new unmaintained component F. In order to avoid the infinite regress of adding further and further new maintenance elements, we add just one more, namely, a process that maps B to F implemented by $\mathtt{f}$ itself ($\mathtt{f} : B \to F$). (For a discussion of the specific conditions under which this is possible, the reader is referred to Rosen's original argument [23]; alternatively, Wolkenhauer [26] offers a very clear discussion.) Now every component (except for the presumed input A) is maintained in the system, and every mapping is implemented by a component.

4.1 The Proof

In this subsection we will recap Rosen's proof that living systems are not implementable in computational systems, or more precisely, that the relational model in Figure 7 cannot be reconstructed from a state description. Thus, the system contains more than can possibly be encoded in a state description—it is not implementable in a computational system. In what follows we will closely follow Rosen's own exhibition of the proof in Chap. 9 of *Life itself*.

Let us assume that the system in Figure 7 is indeed a mechanism, that is, it is completely specified in terms of a set of states. Due to the equivalence of synthetic and analytic models of mechanisms, every mechanism can be considered as a direct sum of its components. We can therefore assume that its components $\mathtt{f}$, F, B, and A can also be completely described in terms of states. In such a way one gets models for each of the components in terms of states. Given those components, one can then generate a synthetic model of the entire system with the models of the components as summands. If the system in Figure 7 is a mechanism, then this synthetic model should be a complete description of the model.

Rosen argues that this is not possible, however: Let us assume that there is a most refined model of $\mathtt{f}$ in terms of states that we can write as $\{\mathtt{f}_1,\ldots,\mathtt{f}_k\}$. We would now like to know whether it is possible to recover the original component $\mathtt{f}$ from this state description; more specifically,

whether we can recover the functional overloading of $\mathtt{f}$ (that is, $\mathtt{f}$ maintains both $B \to F$ and $A \to B$).

Let us consider two possibilities: The first possibility is that the first $j_0 < k$ components of $\mathtt{f}$ implement the mapping $A \to B$ and the remaining components implement $B \to F$. In this case the component $\mathtt{f}$ is of the form $\mathtt{f}_{A\to B} \oplus \mathtt{f}_{B\to F}$. If this is so, then in order to retain the closure of the system we need to have each of the components $\mathtt{f}_{A\to B}$ and $\mathtt{f}_{B\to F}$ itself be maintained by something within the system. This means that we need to assume the mapping $B \to \mathtt{f}$ to be in fact two mappings, namely $B \to \mathtt{f}_{A\to B}$ and $B \to \mathtt{f}_{B\to F}$. This in turn means that we need to split up the component F into two parts as well: one that implements $B \to \mathtt{f}_{A\to B}$ and one that implements $B \to \mathtt{f}_{B\to F}$. Unfortunately, in this case we will have at least one component of F unmaintained. Continuing the argument will lead to no closure. We therefore need to reject the possibility that $\mathtt{f}$ can be split up into two parts as indicated above.

The second possibility is that even in the most refined model there are some components that implement two mappings. According to Rosen this possibility is not compatible with the assumption of a machine. In the context of machines, where synthetic models are equivalent to analytic models, one would expect that "functions are always localized into corresponding organs" [23, p. 212]. Rosen concludes from this that we must suppose $\mathtt{f} = \mathtt{f}_{A\to B} \oplus \mathtt{f}_{B\to F}$ in the sense of a direct sum [23, p. 240]. This means that if $\mathtt{f}$ implements two functions, then it can be split up into at least two independent components. We are thus back at the first case.

Altogether, Rosen concludes that the system in Figure 6 is not compatible with a state-space-based description and is thus not a mechanism. This proof is the crucial piece in *Life itself*, substantiating the main point that there are systems that they bear the organizational characteristic of living systems (closure). These systems cannot be completely described by a state-space approach. If correct, this would indeed have important consequences for our understanding of life in general and attempts to create artificial life in silico. The implication of Rosen's argument would be that the latter project is infeasible on principle grounds.

5 The Concept of Implementation

But is it correct? Does Rosen's central proof hold? In this section we will show that it does not.

There are two main possible criticisms of Rosen's proof. Firstly, one might wish to question whether Rosen's closure property really represents a necessary feature of living systems. This is an interesting question, but we will not pursue it in the context of this article. The second possible criticism is directed against the proof itself. In this section we will follow this second strategy.

A crucial element of the entire proof is the requirement that implementations of the mappings be localized in specific components of the system. Ultimately it was this requirement that led Rosen to the conclusion that state-based systems cannot realize the closure property of living systems. This requirement of localization says that in machines we can at any time precisely specify which parts of the system implement a particular mapping and which parts do not. Stated contrapositively: Given two components of a system, $\mathtt{f}_1$ and $\mathtt{f}_2$, if neither of those implements a mapping, the compound system $\mathtt{f}_1 \oplus \mathtt{f}_2$ will not do so either. If it did, then we could not determine where precisely the implementation is located. This requirement reflects the idea that machines can be built from parts and are completely determined by their parts. Unfortunately, the localization requirement is unrealistic, and the concept of implementation is ill defined. In the next few paragraphs we will explain this in more detail.

Let us start by reviewing the basic properties of a machine *sensu* Rosen.

- They can be analyzed into their parts.
- The parts themselves can be reassembled to recover the whole system.
- Mappings in machines are implemented by localized parts of the machine.

Those are only the properties of mechanisms as they follow from the equivalence of synthetic and analytic models plus a formulation of the localization hypothesis. One way to understand Rosen's central proof is as saying that those three properties are not compatible with the property of closure. While that may be true, we will now show that even in machines those properties are not compatible.

Consider the component $\mathtt{f}$ in the machine in Figure 6. There $\mathtt{f}$ implements the mapping $A \to B$; thus the mapping $A \to B$ is located in $\mathtt{f}$. So far, no problem. Let us now try to investigate the component $\mathtt{f}$ in more detail and analyze how precisely it implements the mapping. Assume that the component $\mathtt{f}$ is the direct sum of components, $\mathtt{f} = \oplus_i \mathtt{f}_i$. Then according to the localization hypothesis for mechanisms, at least one of the $\mathtt{f}_i$, say $\mathtt{f}_{i_0}$, implements $A \to B$ (or parts of it). It might then be that $\mathtt{f}_{i_0}$ is itself a direct sum; without restricting the generality of the argument, we will ignore this possibility and assume that $\mathtt{f}_{i_0}$ is an atomic component. Due to $\mathtt{f}$ being a mechanism, we know that the component is exhaustively described by one state only; call the state variable $x_{\mathtt{f}_{i_0}}$; if we needed two or more states to describe it, then $\mathtt{f}_{i_0}$ would not be an atomic component and we could simply split it up further and further until we did find an atomic component.

The problem of this becomes clear when one attempts to recover information about the mapping implemented by the component $\mathtt{f}_{i_0}$ if this component is described in terms of the $x_{\mathtt{f}_{i_0}}$. By the same token, given a set of states, how can one decide whether or not they actually do implement a mapping or not? Our conclusion is that one cannot. To see this, consider an implementation of the system in Figure 6. Specifically, assume an implementation of this system in an N-dimensional cellular automaton (CA). By the assumption of mechanism, such a CA is guaranteed to exist. Moreover, because of the finiteness of mechanisms, there will be a minimum required size for the CA to implement the system. Let us assume that K is such a CA. Assume that K implements the machine in Figure 6. Furthermore assume that K' is a copy of K except that we assume all its cells are in the quiescent state. Given those systems, we can now form a new system G by taking the direct sum $K \oplus K'$.

The system G is now completely specified by a set of states:

$$\underbrace{k_1, k_2, \ldots, k_n}_{\text{system } K}, \underbrace{k'_1, \ldots, k'_n}_{\text{system } K'}$$

Given this complete model of the machine, we will now assume that we can actually identify the state variable $x_{\mathtt{f}_{i_0}}$ with a state variable of K, say k_l. We conclude that the state variable k_l implements a mapping.

Let us now assume a slightly different scenario. Specifically, assume that the system in Figure 6 is implemented in the part of G that corresponds to the summand K', while we assume now that all the cells of K are kept in a quiescent state. Now, the state variable k_l still describes some state of K, but it does not describe a mapping any more; this is so because we assumed that K is an empty system (all cells in the quiescent state). On the other hand, we would now have to assume that the corresponding state k'_l of the subsystem K' does implement a mapping.

This example thus seems to indicate that whether or not a state implements a function does depend on the context of the system as a whole. This however is in clear contradiction to the requirement that the state description be a complete description.

We are thus left with two possibilities. The first possibility is that we acknowledge that machines are not compatible with the localization hypothesis. In this case Rosen's proof is rendered irrelevant, because all he has shown is that closed systems share this property with machines; this does not tell us whether or not closed systems are mechanisms; thus we still do not know whether artificial life in silico is possible.

The second possibility is that Figure 6 is indeed not a valid model of a machine because it is incompatible with the localization hypothesis. This might well be the case, but would render the

proof irrelevant in that we have only shown that the systems in Figure 6 and Figure 7 do not share a certain property. Since the system in Figure 6 is a machine, all we know now is that neither machines nor closed systems have this localization property. This again does not tell us anything about the difference between machines and closed systems.

Altogether, it seems that Rosen's proof does not hold and we are still entitled to conclude that organisms might be machines.

5.1 Syntax and Semantics

In this section we will revisit the notion of implementation of a mapping. This notion has been central to the idea of a relational model of a machine and thus ultimately to the definition of closure. The discussion in the last section indicated already that there is a problem with the notion of the implementation of a mapping. Particularly, it seems unclear how one can recover the relational models of machines from their state-based models.

Rosen's own comments on the nature of the mappings and their implementations in relational models are somewhat unclear. On the one hand, it seems that the localization hypothesis suggests that a mapping is a syntactic component of the machine, in the sense that there is an algorithm that allows one to specify all mappings in the machine, given a complete description of the machine. On the other hand, though, there are passages in *Life itself* that seem to suggest otherwise. For example, referring to components that implement mapping Rosen writes

> [...] its description changes as the system to which it belongs changes. It can thus *acquire* new properties from the larger system with which it associates. [23, p. 121]

The ability of a component to acquire new properties from the environment is itself not compatible with the idea that it is a mechanism, that is, with the idea that the component is determined entirely in terms of its states. From the above quotation it seems that the concept of a mapping and its implementation are semantic; this does not fit very well with the idea of a mechanism as an inherently syntactic system (syntactic in the sense that all aspects of the machine can be derived from the complete description of its parts). If the description of the mechanisms changes depending on the context of the machine, then the description obviously cannot be recovered from a complete description of the machine itself. On the other hand, if the relational models of a machine can be derived from the state description, then Rosen should have specified a suitable algorithm. He has not done so.

A concept that springs to one's mind when reading Rosen's statement above is that of *emergence*. Particularly in various bottom-up models, it is often observed that the interactions between components lead to new and surprising aggregate effects. It is in this context that one often observes that components acquire new properties that also might change according to the context. Of course, the actual properties of the components do not change, and often they cannot change because they are "hard-wired." One might for example think of interacting agents in agent-based models. What does change is the meaning of the component and its relation to other components in its environment. Thus the emergent effects we observe in bottom-up models are of semantic nature; something is emergent because we think it is interesting.

Altogether, it seems that relational models are semantic models in the sense that the relation between components in those models cannot be derived from the a complete description of the system in terms of states. Relational models might be valuable in certain contexts to describe both machines and organisms, yet in the context of Rosen's "proof" they are misplaced. In fact, by representing machines as relational models, Rosen already leaves the realm of mechanisms. Even more, relational models of living systems, or indeed any other relational models, cannot be mechanisms, due to their semantic nature.

6 Discussion and Conclusion

This article contains two main contributions. Firstly, a clarification of the notions underlying Rosen's central proof. We have mainly concentrated on a comprehensive outline of the mathematical concepts to clarify the mathematical background of the most important notions underlying Rosen's work. A detailed mathematical and formal treatment including formal definitions and proofs will be provided elsewhere.

Secondly, this article also contains a critical examination of Rosen's central theorem. The main conclusion of this is that relational models rely on an understanding of the system and its context and function, while state-space-based models only reflect purely formal aspects of systems. By its very definition such a description cannot per se contain any semantic aspects. Biologically important notions, such as "function" (an inherently semantic concept) can therefore trivially never be recovered from such descriptions. Functional (*sensu* biology) aspects of systems are invisible in a state-space-based description of systems. Yet, this does not mean that there are no functional components in the system. Think for example of the organs in von Neumann's self-reproducing machine.

Where does that leave attempts to create artificial life in silico? In the absence of a generally accepted example of a living system in silico, we cannot know for sure that computational processes can implement life. At the same time, Rosen's argument certainly does not show that they cannot. Rosen's published works do not contain a final answer to the fundamental question of artificial life: Is life in silico possible? Nor does it contain a final definition of life. Despite this shortcoming, we still think that Rosen has something to contribute. His work reflects on the status of models, and on the relations between syntax and semantics and between reductionism and emergence. Those topics are still poorly understood, although of high relevance for artificial life research. We believe that Rosen's circles of ideas contain valuable contributions to those big questions, and their reevaluation will help to bring us closer to an understanding of life and artificial life.

Acknowledgments

We thank Roger Strand and Noel Murphy for discussions on the topic, and Barry McMullin for comments on the manuscript. D.C. thanks the Norwegian Research Council for financial support.

References

1. Adamatzky, A. (Ed.) (2002). *Collision-based computing.* Cambridge: Springer-Verlag Telos.
2. Adami, C. (1998). *Introduction to artificial life.* New York: Springer-Verlag.
3. McMullin, B. (1997). *Scl: An artificial chemistry in swarm* (Technical report bmcm9702). Dublin City University, School of Electronic Engineering; working paper 97-01-002, Santa Fe Institute.
4. McMullin, B. (2000). Some remarks on autocatalysis and autopoiesis. *Annals of the New York Academy of Sciences, 901,* 163–174. http://www.eeng.dcu.ie/~alife/bmcm9901/.
5. Bedau, M., McCaskill, J., Packard, P., Rasmussen, S., Green, D., Ikegami, T., Kaneko, K., & Ray, T. (2000). Open problems in artificial life. *Artificial Life*, *6*(4), 363–376.
6. Burks, A. W. (Ed.) (2002). *Theory of self-reproducing automata [by] John von Neumann.* Urbana: University of Illinois Press.
7. Casti, J. (1992). *Reality rules: I The fundamentals.* New York: John Wiley & Sons.
8. Casti, J. (1992). *Reality Rules: II The frontier.* New York: John Wiley & Sons.
9. Casti, J. (2002). Biologizing control theory: How to make a control system come alive. *Complexity*, *7*(1), 10–13.
10. Lawvere, F., & Schanuel, S. (1997). *Conceptual mathematics.* Cambridge: Cambridge University Press.
11. Letelier, J., Marin, G., & Mpodozis, J. (2003). Autopoietic and (M,R) systems. *Journal of Theoretical Biology*, *222*(2), 261–272.

12. Maturana, H., & Varela, F. (1980). *Autopoiesis and cognition: The realization of the living*. Dordrecht: Reidel.

13. McMullin, B. (2000). John von Neumann and the evolutionary growth of complexity: Looking backwards, looking forwards. *Artificial Life*, *6*(4), 347–361.

14. McMullin, B., & Groß, D. (2001). Towards the implementation of evolving autopoietic artificial agents. In *Proceedings of the 6th European Conference on Advances in Artificial Life* (pp. 440–443). Springer-Verlag.

15. McMullin, B., & Varela, F. (1997). Rediscovering computational autopoiesis. In P. Husbands & I. Harvey, (Eds.), *Proceedings of the Fourth European Conference on Artificial Life*. Cambridge, MA: MIT Press. http://www.eeng.dcu.ie/~alife/bmcm-ecal97/.

16. Nomura, T. (2003). Formal description of autopoiesis for analytic models of life and social systems. In *Proceedings of the Eighth International Conference on Artificial Life* (pp. 15–18). Cambridge, MA: MIT Press.

17. Ono, N., & Ikegami, T. (1999). Model of self-replicating cell capable of self-maintenance. In D. Floreano, J. Nicoud, & F. Mondada, (Eds.), *Proceedings of the 5th European Conference on Artificial Life* (*ECAL99*) (pp. 399–406).

18. Ono, N., & Ikegami, T. (2000). Self-maintenance and self-reproduction in an abstract cell model. *Journal of Theoretical Biology*, *206*, 243–253.

19. Pierce, B. (1991). *Basic category theory for computer scientists*. Cambridge, MA: MIT Press.

20. Rasmussen, S., Baas, N., Mayer, B., Nilsson, M., & Oleson, M. (2001). Ansatz for dynamical hierarchies. *Artificial Life*, *7*(4), 329–354.

21. Ray, T. (1996). *An approach to the syntheses of life* (pp. 111–145). Oxford: Oxford Univesity Press.

22. Rosen, R. (1978). *Fundamentals of measurement and representation of natural systems*. New York: Elsevier North-Holland.

23. Rosen, R. (1991). *Life itself*. New York: Columbia University Press.

24. Rosen, R. (1999). *Essays on life itself*. New York: Columbia University Press.

25. Varela, F., Maturana, H., & Uribe, R. (1974). Autopoiesis: The organisation of living systems, its characterization and a model. *Biosystems*, *5*, 187–196.

26. Wolkenhauer, O. (2002). Systems biology: The reincarnation of systems theory applied to biology? *Briefings in Bioinformatics*, *2*(3), 258–270.

AUTHOR QUERY

AUTHOR PLEASE ANSWER QUERY

During the preparation of your manuscript, the question listed below arose. Kindly supply the necessary information.

1. Please confirm if the proposed running head is okay.

END OF QUERY