Walter Fontana

Systems Biology
Harvard Medical School, Boston, MA 02115
wwww.walterfontana.science

Du calcul au vivant: le défi d’une science de l’organisation

Leçon inaugurale, chaire Informatique et sciences numériques 2019-2020
Collège de France, Paris
24 octobre 2019

From computation to life: The challenge of a science of organization

Inaugural Lecture, Chair in Informatics and Computational Sciences 2019-2020
Collège de France, Paris
October 24, 2019

Version française

**Published as Leçons inaugurales du Collège de France. no 291. Éditions Fayard.
ISBN: 978-2-213-71683-1 (2020)**

English version

Mr. Chairman of the Collège de France,
Mr. President of Inria,
Mr. President of CNRS,
Dear colleagues, dear friends,
Ladies, gentlemen,

I would like to begin with thanking professor Gérard Berry and the professors who nominated me to the annual chair in Informatics and Computational Sciences. I also wish to thank the assembly of professors of the Collège de France who made the decision to invite me for this academic year. I am deeply honored and grateful. My gratitude also extends to INRIA who created this position in partnerhsip with the Collège de France and supports it financially. Finally, many thanks to professor Xavier Leroy for his substantive introduction.

Introduction

Computer science and biology have many points of contact. To orient my lecture, I briefly outline these connections along an axis of increasing conceptual entanglement.

(1) Scientists collect large amounts of data about structures, sequences and states of biological molecules. These data are annotated and organized with the help of a computational "library science" that builds digital warehouses and designs *knowledge representations*.

(2) The accumulated data are scanned for patterns using methods from statistical inference. Statistical inference is being reshaped by *machine learning*, or artificial intelligence, which is a powerful approach originating in computer science.

(3) In addition to statistical models, researchers also construct *mechanistic models* to gain insight into the dynamical processes that generate the system state reflected in the data. Analyzing the behavior of a molecular interaction network is helpful for understanding how and why a biological system might function. Such networks are modeled at various levels of abstraction. One recent approach represents each interaction as an instruction in a purpose-made programming language. A model then effectively represents a biological system as a *program*. This is more subtle than just using a computer; it is about *representing* a complex system using ideas from computation.

(4) At a more fundamental level, many systems in nature are composed of components that mutually construct each other in a way that glues them together into a unit: metabolisms, cells, organisms, ecologies, cognitive systems, economies, cultures. All these systems are *functional organizations*. What kind of dynamics produces organizations of this sort? How much of their architecture is contingent and how much of it is inevitable? The *idea* of computation is the modern formalization of the idea of mechanism. However, unlike its predecessors, the clockwork and the steam engine, computation emphasizes a constructive aspect of interaction. Is this notion of mechanism fundamental to our understanding of nature?

In my lecture tonight I will emphasize modeling, point (3), and barely mention databases and statistical inference, points (1) and (2), although they are of critical importance to modeling. However, I will seek to approach my subject through point (4), the foundations. To risk something that might fail is the least I can do to thank you for the invitation to occupy this chair.

My lecture has one thread: The *idea* of chemistry, which includes, of course, the organic chemistry we know, but also the evolved chemistry that operates through proteins and organizes the behavior of living cells. I will start, however, by viewing computation as a "chemistry" by defining a sort of "ideal chemistry of logic". This chemistry gives rise to organizations that are not the typical subject of physics and perhaps not even of traditional computer science, yet they exhibit a phenomenology that we encounter in living systems. In this way, I will set the motivation for the kind of questions I wish to ask with more faithful, and thus practical, representations of organic chemistry and molecular biology to which I turn subsequently.

Two remarks before I proceed: First, French is neither my native language nor my working language. In fact, this is my first-ever lecture in French. Never mind the circumstance. What I'm doing here is the linguistic equivalent of a BASE jump. I owe a great deal of gratitude to the person who helped repair a 40-year long neglect of my high-school French: Athie Tschibelu, my French instructor in Boston. My second remark is that this lecture is not an overview of my course. Rather, I felt it is more appropriate for this occasion to expose a more philosophical and personal vision of the subject.

The chemistry of computation

I now proceed to violate most physical sense by turning to the theoretical foundations of computation. The chemistry of computation is fairly obvious: $5 + 3 = 8$ is a "chemical" reaction. The number 5 reacts with the number 3 in a mechanism we call plus to construct the number 8. But what is a mechanism of the plus-interaction? For this we need to descend a level deeper into a different world where there is no distinction between numbers and operations; a level at which only objects exist that can act on other objects to produce new objects. This is a world in which everything is defined in terms of behavior. It is here where the foundations of computation live. Alan Turing's work showed that there is no conceptual distinction between programs and data. Data can be programs that programs act upon to produce data which in turn are programs. This sounds very chemical. There is another foundation of computation, equivalent to Turing's, in which the interchangeability of data and programs is completely natural and made syntactically explicit. It is a world of functions in which a function f acts on another function g given to it as an argument to yield another function h as the value. These functions are expressed as symbolic strings in a particular grammar, whose details are not important here. Two strings interact by concatenation (with a bit of syntactic glue), which then triggers a rewriting process that follows just one law. When the process stops we are in possession of an expression that represents the new function resulting from the application of f to g. That's it. In the following, this framework, known to computer scientists as λ-calculus, serves as a completely self-contained universe of interacting objects, regulated by a universal law. If this were the world, it would be a "theory of everything".

Now that we have a chemistry, we are ready for an experiment in which we mix this calculus with a tiny bit of worldliness. First, we are not interested so much in the behavior of single functions, but in the behavior of an ensemble of functions—much like what is happening today in systems biology. Second, like in real chemistry, we will allow any given function to occur in multiple copies; a function, thus, has an abundance or concentration. You now may think of functions as particles[1].

[1] W. Fontana and L. W. Buss, "The barrier of objects: from dynamical systems to bounded organizations", in J. Casti and A. Karlqvist (dir.), *Boundaries and Barriers: On The Limits To Scientific Knowledge*, Reading (Mass.), Addison-Wesley, 1996, p.56-116. W. Fontana and L. W. Buss, "What would be conserved if "the tape were played twice"?", *Proceedings of the National Academy of Sciences of the United States of America*, 91/2, 1994, p.757-761.

We start by filling a flask with, say, 1000 randomly generated particles and impose two dynamical laws. The first law is “making”: pick two particles from the mixture at random, call them f and g. Let them interact as $f(g)$ to construct a new particle h. Keep f and g in the mixture and add h. Now we have 1001 particles. The second law is “forgetting”: pick a particle from the soup at random and remove it. Now we are back to 1000. Repeat forever. The second law means that no particle lasts forever: eventually it will be picked and removed.

Initially, interactions produce many new types of particles; but let us fast-forward and think long-term. Consider some particle h. If h is not the product of any interaction, it will sooner or later disappear because of the *forgetting*-law. So let us imagine h to be the product of the interaction between f and g. If either f or g disappear, h can no longer be produced in that way. If this is the only way, then h is doomed. The problem then shifts to f and g. For f and g not to disappear they must in turn be products of some interaction. And so on. Hence, h will persist in the system only if there is a *constructive* feed-back loop. At this point the system has become collectively self-maintaining. Keep in mind that these particles are informational particles, they are not physical. This is not material self-maintenance, this is logical self-maintenance. I call such a system an *organization*.

We can analyze such an organization the way a chemist would who has no working knowledge of quantum mechanics—in this case λ-calculus. The analysis will reveal some interesting properties that go hand-in-hand with self-maintenance. I illustrate them with the simplest example (Fig. 1), but they hold for much more complex cases.

1. The objects belonging to an organization can be decomposed into building blocks and described by means of a *grammar*. This grammar is not the same that defines the objects at the microscopic level of λ-calculus. Rather, it is a language specific to that organization. Likewise, all interactions can be described in terms of rules that define the behavior of these building blocks. This means a self-maintaining organization admits a coarse-grained description that is independent of the underlying microscopic mechanics of λ-calculus. However, in this change of description, we had to replace a universal microscopic law of interaction with a set of more specialized macroscopic *rules*.

2. The system is algebraically *closed* in that objects within the organization interact to produce only objects within the same organization. A self-maintaining system is a syntactical and behavioral unit without requiring a physical enclosure.

3. An organization typically consists of an infinity of objects, but only a finite core is persistently maintained in a small reaction volume that allows for only 1000 particles. This requires *kinetic confinement*, which has to do with the number of ways in which an object can be produced within the organization. An organization has a center and many more roads lead toward it than away from it.

4. A fourth property is *constructive stability*. Imagine an interaction network in which an object of type A acts on an object of type B to produce a copy of the latter, another instance of the B-object; likewise, a B-object acts on an object of type C to copy the C-object and so forth until the system closes on itself. This is a self-maintaining system, but if you remove one object type completely, say the A-type, the whole system collapses. In contrast, consider an A-object that interacts with another instance of type A to produce an object of type B, and

the B-object acts on another instance of type B to produce an object of type C, etc. Now we can remove all but one arbitrary object type and the system will regenerate itself completely.

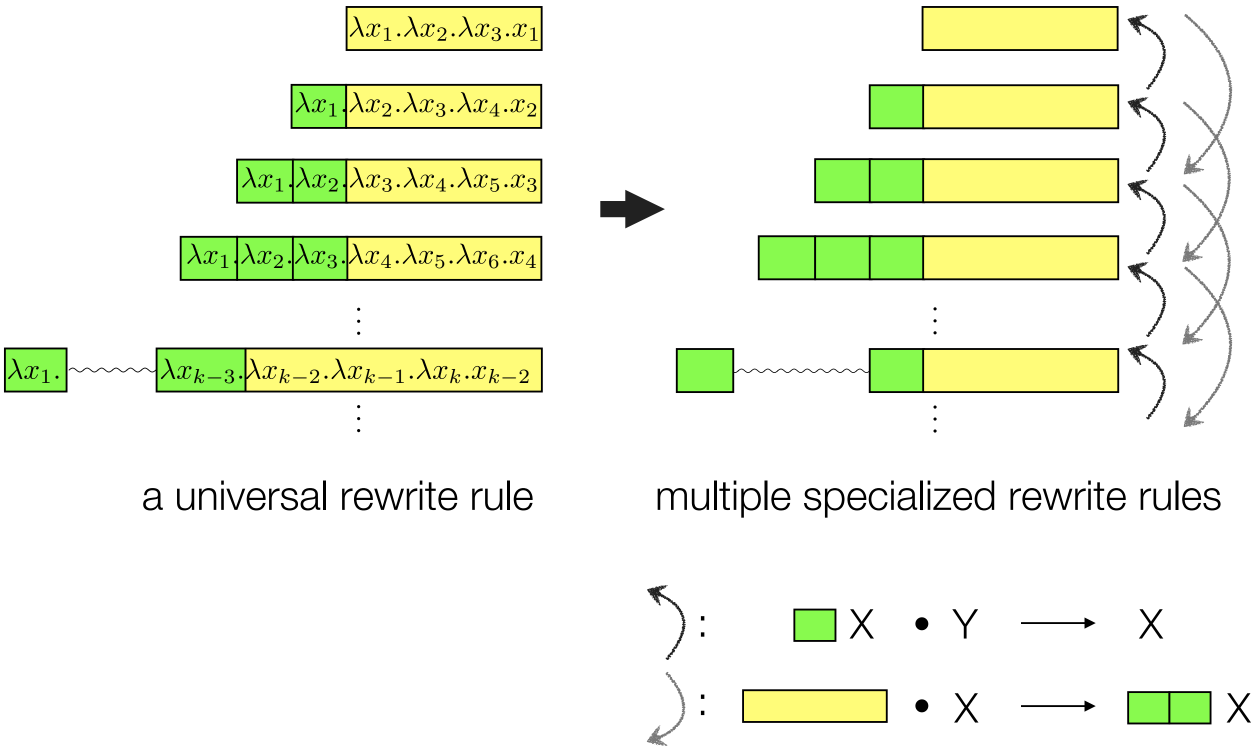

$\lambda x_1.\lambda x_2.\lambda x_3.x_1$
$\lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.x_2$
$\lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.\lambda x_5.x_3$
$\lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.\lambda x_5.\lambda x_6.x_4$
$\lambda x_1.$ $\lambda x_{k-3}.\lambda x_{k-2}.\lambda x_{k-1}.\lambda x_k.x_{k-2}$
a universal rewrite rule
multiple specialized rewrite rules
: X • Y → X
: • X → X

Figure 1: A system that has become self-maintaining can be described by an emergent grammar which abstracts the underlying microscopic structure.

There is one further point of interest. We can *perturb* an organization by presenting it with an object that exists in the full universe but is not in the language of the organization. Consider an organization made of particles that conform with the "yellow" language (Fig. 2). Let "blue" stand for anything that is not yellow. A blue particle then interacts with yellow particles to produce a lot of blue noise. Sooner or later the blue particle disappears and with it all the blue noise. But if interactions within the blue noise, or between it and the organization, regenerate the perturbing particle, we get a new self-maintaining organization that extends the old one by a layer. We can also perturb an organization with another organization. The two organizations can become integrated if their interaction produces a cloud of byproducts that is *not* self-maintaining on its own: a "glue". The point is that organizations can change. Yet, their change is highly constrained.

At about the time Leo Buss, an evolutionary biologist, and I played with λ-calculus to capture the idea of a chemistry, Gérard Berry and Gérard Boudol[2], inspired by work of Jean-Pierre Banâtre and Daniel Le Métayer, played with the idea of chemistry to capture a new form of computation now known as concurrency.

Two insights from the ideal chemistry of logic provide us with a transition to organic chemistry and molecular biology.

The first is catalysis. In chemistry, a catalyst facilitates a transformation without being consumed in the process. We often associate catalytic function with specific individual chemical or physical

[2] G. Berry and G. Boudol, "The chemical abstract machine", *Theoretical Computer Science*, 96/1, 1992, p.217-248.

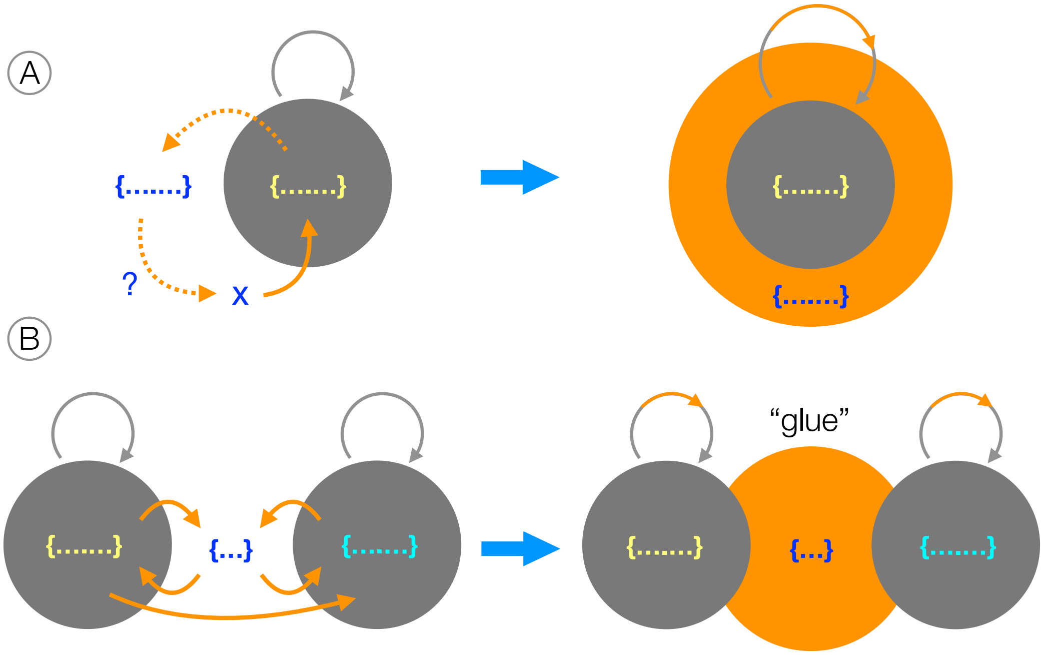


Figure 2: (A) An organization is closed under interactions of its members, which are characterized by a grammar and specific rules of interaction (see Fig. 1). Such an organization can be perturbed by an object of the host universe that does not belong to it. (B) Same situation as in A, but instead of a single object, a whole organization perturbs another.

objects. In the chemistry of logic, catalysis was imposed at one level: f and g were not used up in the production of h. Yet, the requirement for persistence of h led to a different embodiment of catalysis: network catalysis. Fig. 3 illustrates the concept from a more chemical perspective in which reactants are consumed directly by the reaction rather than by random forgetting as before. Here A reacts with a to produce B and a', B reacts with b to produce C and b', and so on until A is regenerated. Regardless of whether each reaction is catalyzed by some other agent, the cycle itself acts as a catalyst: The members of the cycle $\{A, B, C, D\}$ collectively transform the inputs $\{a, b, c, d\}$ into the outputs $\{a', b', c', d'\}$, while remaining unchanged in the overall balance. Catalytic cycles of this kind are responsible for self-maintenance in our toy universe and in biology. A particularly important case arises when at least one of the outputs is a member of the catalytic cycle. Such a cycle makes more of itself at every turn. This is known as network auto-catalysis.

The second insight pertains to scales of resolution. Picture an axis that indicates the resolution at which we describe interacting objects. On the extreme right is the case in which objects are treated as black boxes to which we assign proper names so we can at least distinguish one from the other. Call one such object "Aspirin". Because there is no structure we could refer to in defining interactions, we need to explicitly list each and every interaction associated with "Aspirin". As we find more, the list becomes longer and longer. On the extreme left we know objects at maximum resolution; say we know their quantum wave function. In this realm, the interactions between objects follow from a general physical principle, like a principle of least action. It is clear that a wall of complexity makes the two extremes intractable in all but the simplest situations. On the left, the *size* of your equations explodes; on the right, the *number* of your equations explodes. In between these two extremes there is a vast territory in which something but not everything about the structure of objects is exposed. This means we can specify interactions by reference to at least

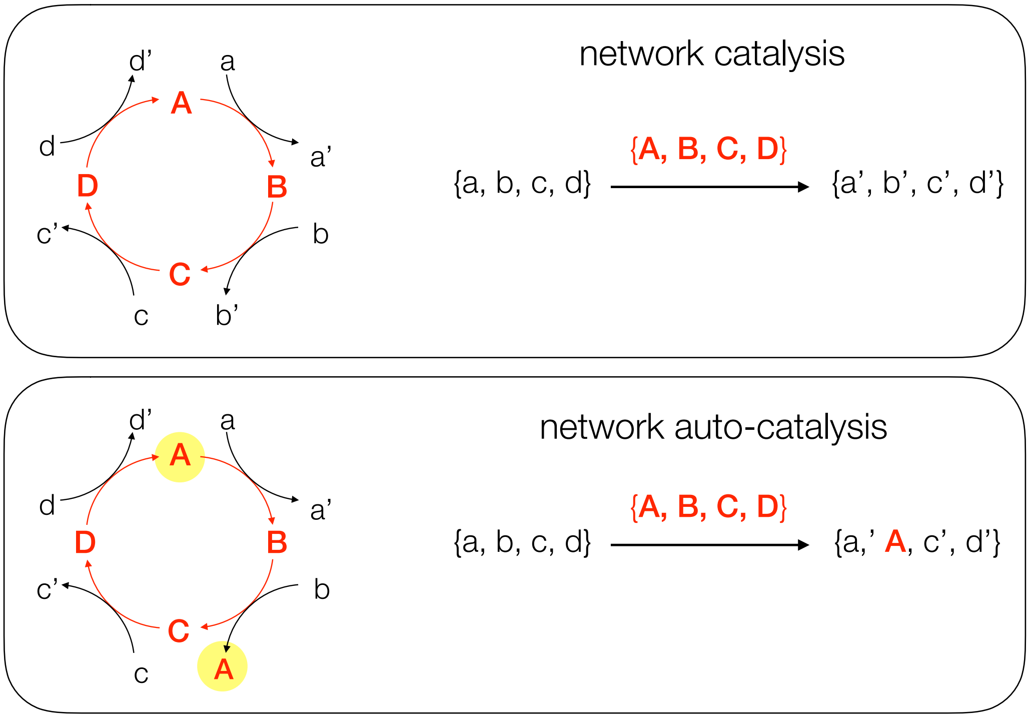


Figure 3: Network catalysis and network auto-catalysis.

some structure, as is the case with the rules of chemistry to which we turn shortly. In this vast land of the middle, the number of rules that govern interaction will be small compared to an explicit enumeration of all possibilities but large compared to a theory of everything. Our excursion with the ideal chemistry of logic suggests that this is the land of functional organization, which is to say, the land of tunable abstraction.

Organic chemistry

Today, organic chemistry is anchored in quantum physics. Yet, chemistry enjoys considerable autonomy from quantum physics. In fact, chemistry has developed its own language, to a good measure thanks to Lavoisier, who was inspired by a powerful idea—due to Étienne Bonnot, Abbé de Condillac—that “languages are true analytical methods”. This could be a one-line characterization of computer science some two hundred years before it existed. Today the language of chemistry is largely that of graphs.

A molecule is a structure that electrons have shaped to be comfortable. But comfort is ephemeral. Another molecule comes along and with it the possibility of more electronic comfort. The result is a transformation of molecules in a chemical reaction. Because molecules can be represented as graphs, a reaction can be represented mathematically as the rewriting of a graph. This is a well-known formalism in computer science.

A key point of chemistry is the distinction between a *reaction* and a *mechanism*. In a reaction, all molecular parts are specified. Yet, empirically, not every part determines the resulting transformation. We can vary experimentally some parts and they seem to remain untouched in the reaction. This leads to the idea of a reaction mechanism as the specification of only those parts that are necessary for a particular transformation. A mechanism is therefore the transformation of a pattern. A mechanism is a rule, much like the rules of ideal chemistry that we saw before.

A rule can be applied by first checking whether the pattern on its left matches candidate molecules (Fig. 4). If there is a match, then the matched parts are transformed in place as specified by the rule. The match could also occur within a single molecule. At this level of abstraction, a rule represents empirical knowledge on top of first-principles, such as the octet rule, which constrain legal bond structures.

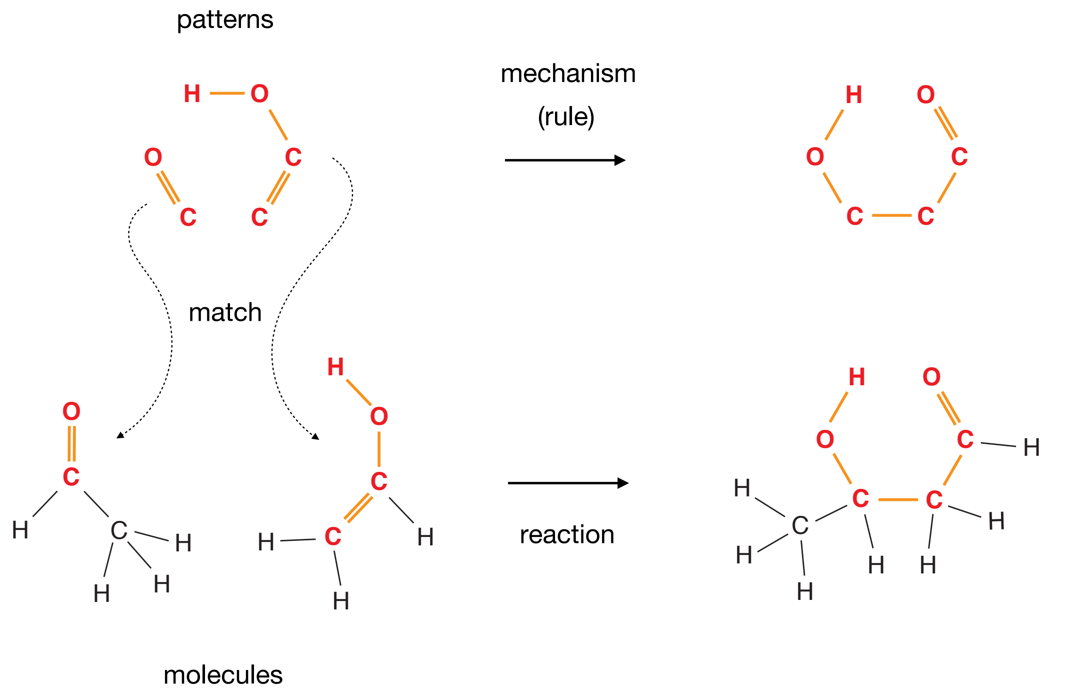


Figure 4: Chemistry distinguishes between the transformation of molecular parts, or patterns, and the resulting reaction between molecules.

Exploring chemical space requires a platform for handling rules. Given the significance of chemistry for life in a biological, technological, and commercial sense, it is surprising that a rigorous and open-source platform for modeling chemistry in terms of graph-rewriting has been implemented only over the past decade. Graph-rewriting is a natural representation and a fertile terrain for formal methods with links to category theory. The platform, called Mød[3], has been conceived an implemented by Daniel Merkle, Jakobe Lykke Andersen, Christoph Flamm, and Peter Stadler.

While a chemical rule formalizes empirical observations, it can also be informed by insights at a lower, more detailed, level of description. A molecule is an electronic arrangement and a reaction is an electronic re-arrangement that can be conceptualized in terms of elementary electronic displacements expressed through symbolic arrows. How arrows should be written is codified by rules based on quantum-physical principles—a codification known as *arrow pushing*. The arrow of a chemical rule can thus be unpacked into a more detailed mechanism at the lower level of arrow pushing. This mechanism need not be strictly sequential, as some electronic displacements can occur independently of others. We might go the other direction and think of a chemical rule as

[3] J. L. Andersen, C. Flamm, D. Merkle et P. F. Stadler, “A software package for chemically inspired graph transformation”, R. Echahed et M. Minas (Eds.), Graph Transformation: 9th International Conference, ICGT 2016, Springer, Lecture Notes in Computer Science Series, vol. 9761, 2016, p. 73-88.

abstracting the underlying arrow-pushing network that describes electronic displacements. In the vast land of rule-based representations we can go up and down levels of abstraction by composing rules. The abstraction level of chemical rules is justified by the stability of the molecules to which rules are applied. In contrast, the states generated by arrow pushing are ephemeral and foremost conceptual.

Obtaining the rules of chemistry is a difficult open problem. Perhaps the largest chemical database is Reaxys, owned by Elsevier. Reaxys extracts data from $16,000$ journals and patents from 1771 to today. On the order of 20 million compounds and 16 million reactions are usable for learning rules. Any approach must overcome many challenges, including lawyers. Some of the difficulties consist in figuring out for every reaction which atoms on the left correspond to which atoms on the right; identifying the necessary context of a transformation which may include parts that are not themselves affected but nonetheless required; estimating energetic feasibility; assessing stereochemistry; identifying reaction conditions, such as type of catalytic support and solvent, as well as temperature and pressure.

It seems perfectly reasonable and feasible to identify rules that pertain to specialized areas of chemical space, such as sugars or fats. However, it is unclear to me whether it makes sense to speak of "the" rules of chemistry in full generality. My reasoning is as follows. The rule-based representation seems to come at the price of a separation between physical objects, i.e. the molecules, and epistemic objects, i.e. the rules. I would like to suggest that this separation between objects and rules may not be absolute, especially as we march toward biology.

In biology we find molecular objects, such as proteins, that are very large and provide specific chemical function, often comprising catalysis. So let us return to catalysis. β-lactamase is an enzyme that destroys antibiotic substances known as β-lactams, which block the synthesis of the bacterial cell wall (Fig. 5A). One of the proposed mechanisms by which the enzyme inactivates the antibiotic involves a series of proton exchanges within its catalytic domain, which allow electrons to flow so as to break the antibiotic ring. After the ring has been broken, the enzyme has to clean up after itself to return to its original state (Fig. 5B). Strictly speaking it is not the identical state, since after the clean-up, the protons are not the original ones. But for chemical purposes all protons are equivalent.

Note in Fig. 5B that catalysis is not a single step, but comprises several steps in a little reaction network: It is network catalysis, much like we encountered in the chemistry of logic, but it now occurs in an environment protected by the protein. We can cast the whole process in terms of rules by keeping only those molecular parts that are necessary at the level of abstraction set by chemistry, Fig. 5C. Given two rules and a reason to believe that they apply in a particular sequence, because one rule produces a state that the other rule depends upon, it seems justifiable to compose them into a single rule[4]. We must be aware, however, that this eliminates the possibility of "cross-talk" or interference from other rules that may be operating concurrently. The imposition of such order may or may not be appropriate depending on the question one asks. For example, composing all rules of glycolysis into a single overall rule would prevent any reasoning about alternative fates of intermediates. Composition of rules is a form of abstraction, much like the composition of electronic displacement arrows mentioned before. Within the catalytic pocket of a protein the risk of outside interference is mitigated and we might as well compose all rules into one overall chemical

[4] J. L. Andersen, C. Flamm, D. Merkle and P. F. Stadler, "Rule composition in graph transformation models of chemical reactions", MATCH. Communications in Mathematical and in Computer Chemistry, 80/3, 2018, p. 661-704.

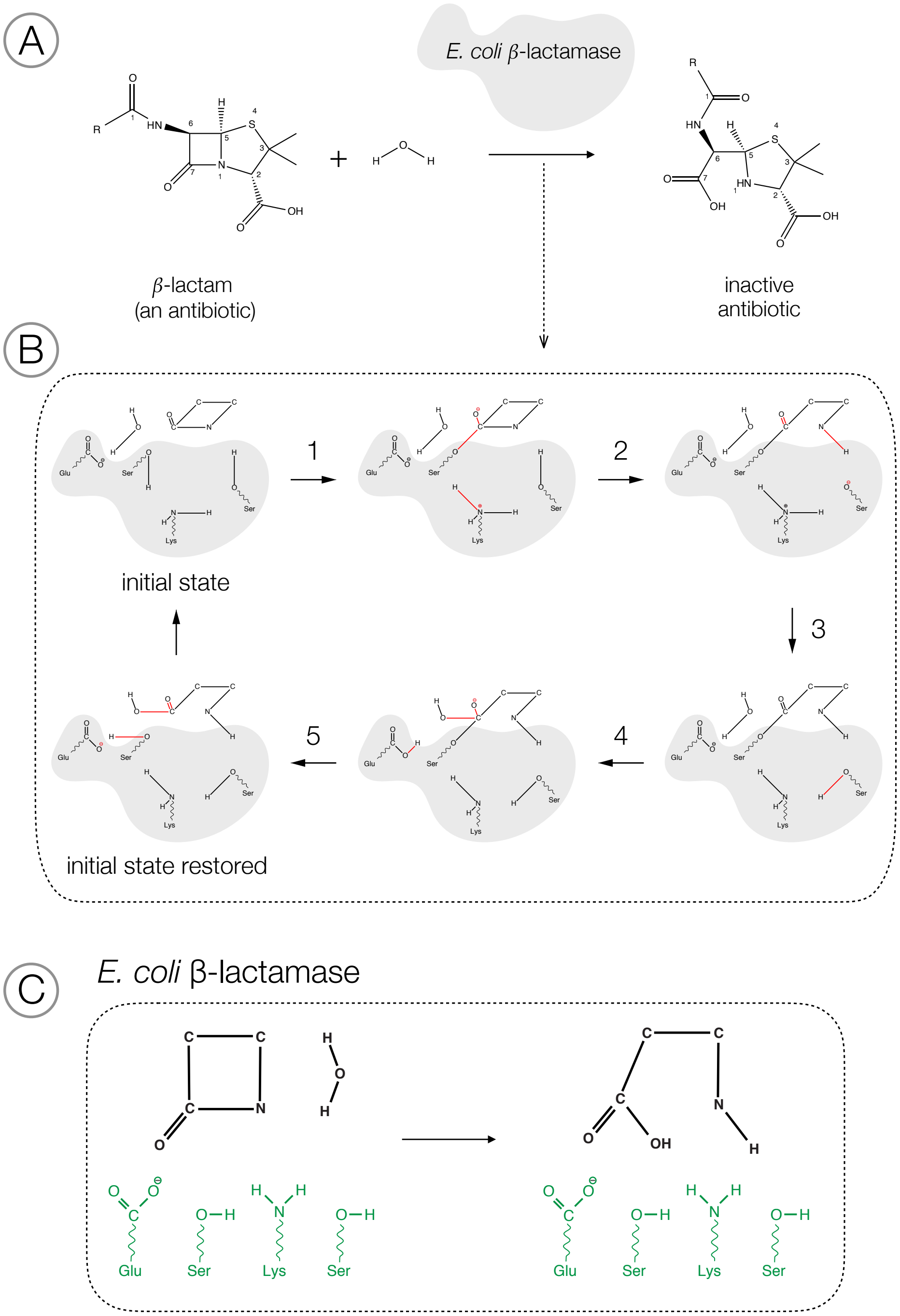


Figure 5: (A) The *β*-lactamase enzyme deactivates an antibiotic. (B) The process occurs in five steps that can be represented by rules. (C) These rules involve amino acid residues that appear as an invariant context of the process by virtue of enabling a catalytic cycle in conjunction with the substrate motif. The composition of the rules into an overall rule is justified because the enzyme guarantees the causal ordering by preventing interference from the outside. Viewed from this angle, the enzyme is a reification of the overall rule that results from this composition. Here, the wiggly lines and amino acid identifiers are just metadata stating that the carboxy, amino, and hydroxy groups come from amino acid residues of the protein.

rule.

Let us do that for β-lactamase. At the bottom of Fig. 5C, I show the overall rule. The green part comes from the enzyme and supplies the context necessary for the transformation. I would like to claim that this rule *is* the enzyme. The black transformation on its own would not occur. The green context makes it possible. The enzyme, however, is not just the green context, but the *whole rule*, since the green context makes no sense without the black pattern. The green context is not a passive context. A passive context would be the carbon-carbon bond that provides connectivity at the top of the lactam ring (in black), but is not altered. In contrast, the green context undergoes a cyclic transformation *only* by virtue of the interaction with the black part. The explicit transformation can be hidden—wrapped—under a single arrow because a single object—the protein—guarantees the causal ordering. This means that, at least in some parts of chemical space, the rules of chemistry can be reified as molecules. However, molecules like proteins are evolvable. We may then think of the rules of chemistry as evolving. Computer scientists call this property reflection. Reflection is possible when a system gains access to the very processes that make it the system it is, such when a programming language provides read/write access to the interpreter that is running it.

On a larger scale we might use the Mød platform to construct networks by repeatedly applying a specific collection of rules to expand an initial set of molecular species and then ask whether the resulting network has become catalytic or autocatalytic.

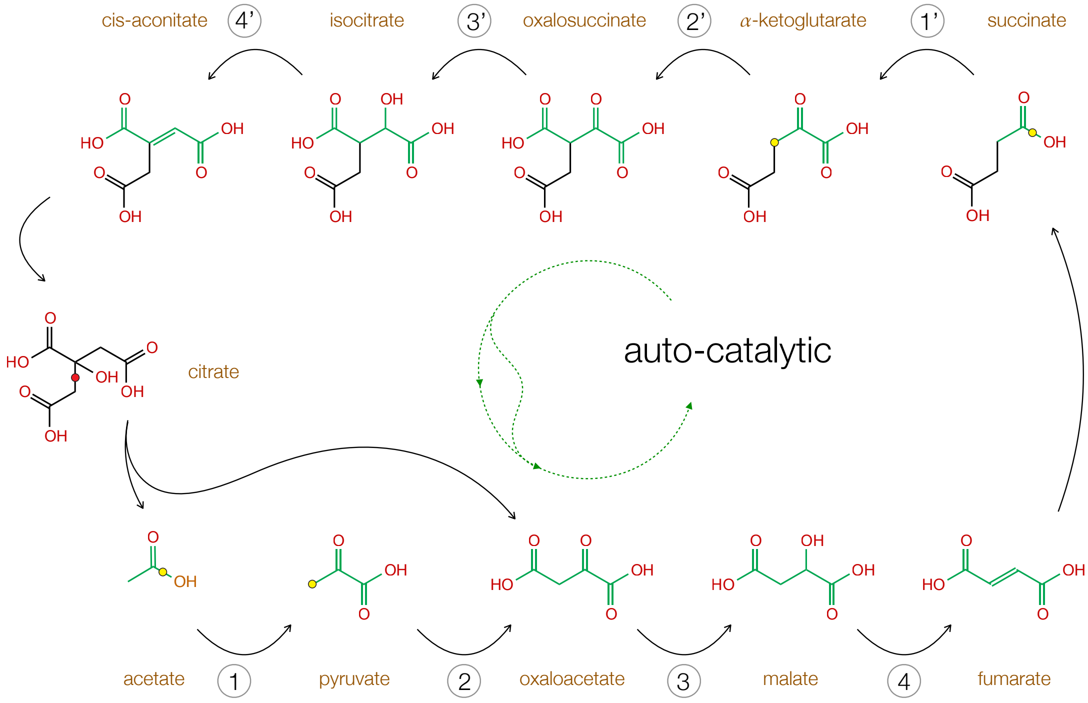


Figure 6: The skeleton of the reductive tricarboxylic acid cycle, showing only its internal molecular components. The yellow dots indicate where the carbon skeleton is extended by CO_2. The red dot indicates where citrate fragments into two members of the cycle, closing it autocatalytically.

An example of autocatalysis that is central to the history of life is the reductive tricarboxylic acid (TCA) cycle, depicted in Fig. 6, which shows only the molecules within the cycle. The cycle running under oxidative conditions is known as the "Krebs cycle", whose chemistry is the core of metabolism in all living systems. It breaks down food stuff and acquires electrons that are transferred to oxygen in the process of respiration, which produces water and energy. Life originated, however, in a reducing atmosphere, well before life itself generated atmospheric oxygen by evolving photosynthesis. In a reducing context, the TCA cycle runs backwards, taking up carbon dioxide and energy to construct materials. In this direction it is autocatalytic. Note its beauty. A molecule grows progressively to reach a symmetric form at which point exactly the same reaction sequence occurs with one half of that molecule, which finally breaks apart to close the cycle while yielding an additional component of the cycle[5].

Auto-catalysis is relevant in origin-of-life scenarios, because it concentrates the mass of a system in the autocatalytic loop, while suppressing combinatorially many side reactions that could be a kinetic threat to the loop. It would be of great interest to understand whether, given hypotheses about the chemical substances and the chemical rules available 3.5–4 billion years ago, this cycle was the only auto-catalytic solution in the accessible chemical space, or whether there is a vast variety of alternative solutions. In other words: is the universality in the functional organization of metabolism that we observe today a "frozen accident"—one of many solutions that just happened to take over—or is it necessary? Eric Smith and the late Harold Morowitz have been asking such questions, and platforms like Mød are needed to move us forward.

It is perhaps surprising that a fully satisfactory *formal* specification and algorithmic detection of auto-catalysis remains challenging. Here is one reason. The overall balance of autocatalysis has to be of the following form: A set of chemicals called Food F and an instance of a target molecule X engage in some network of chemical reactions that produces a set of chemicals we call Waste W and *two* instances of X: $F + X \longrightarrow 2X + W$. Imagine disentangling this overall scheme into three sub-networks: (i) The network through which X unfolds its catalytic action, with net effect $F' + X \longrightarrow X + W'$; (ii) the network that constructs a copy of X, with net effect $F'' \longrightarrow X + W''$; and (iii) the network that links (i) and (ii) by virtue of which X participates, directly or indirectly, in the construction of another copy of itself. However, if the overall net transformation carried out by this connecting network is zero, it can be compressed away, which disconnects the catalytic part from the construction part. In that case X catalyzes the conversion of some Food into Waste but has no bearing on the construction of X from Food, which runs counter the very idea of auto-catalysis. Such a system is not auto-catalytic from a mechanistic standpoint, although the overall balance equation looks auto-catalytic. To identify this condition from a purely graphical perspective is a complex problem[6], and this static perspective does not even consider causal requirements.

The evolved chemistry of signaling

Organic chemistry and network catalysis govern the complex and interwoven routes in which the molecules of life are transformed. A different kind of system detects and processes information that comes in the form of molecules, small and large, whose presence and abundance correlate with specific conditions inside or outside the cell. These molecular signals must be interpreted to elicit appropriate cellular responses, such as repairing, dividing, moving, differentiating, learning, and adapting in real time. A major role in interpreting these signals is played by networks of proteins that affect each other's behavior by labeling each other with chemical markers and forming transient

[5] E. Smith and H. J. Morowitz, "The Origin and Nature of Life on Earth: The Emergence of the Fourth Geosphere", Cambridge, Cambridge University Press, 2016.

[6] J. L. Andersen, C. Flamm, D. Merkle and P. F. Stadler, "Chemical transformation motifs – Modeling pathways as integer hyperflows", IEEE/ACM Transactions on Computational Biology and Bioinformatics, 16/2, 2019, p. 510-523.

complexes.

I hinted before at the role of protein size in catalysis. Size also plays a role in chemical "object identity". When a large object, like a protein, is modified by attaching or removing a small chemical label, it effectively remains the same object even though it has changed chemical composition. We speak of the same protein changing state but not identity. This would be difficult to argue for small molecules. Ethanol is not a different state of ethane. It plainly is a different thing.

A description of this situation warrants a different level of abstraction than organic chemistry. Instead of representing a protein as a chemical substance, we represent it as an agent with sites (Fig. 7). A site is a logical abstraction of whatever physical and chemical aspects underlie a protein's ability to interact in a specific way. This view suggests a *formal* analogy to organic chemistry. A protein is treated syntactically like an atom, and a complex of proteins corresponds to a molecule. Such a correspondence leads to a rule-based approach that follows the same formal idea that we discussed for organic chemistry. A rule asserts the transformation of a graphical pattern. It is applied exactly as in the chemical case by matching its left side to a molecular species (at an abstraction level that considers proteins as agents, see Fig. 7). If there is a match, the transformation is executed as specified by the rule. Several research groups proposed and implemented such an approach independently, but it was Vincent Danos and Cosimo Laneve who gave it a compelling formal foundation from a computer science perspective[7]. This foundation proved critical for establishing theoretical and algorithmic innovations at the core of the so-called Kappa platform developed by Jean Krivine, Jérôme Feret, Pierre Boutillier, Jonathan Laurent and Russ Harmer[8].

Despite using the same graph-rewrite formalism, organic chemistry and protein chemistry commit to different interpretations. In Kappa, interactions occur at sites and sites are situated below the level of agents: The agents own the sites. The language does not provide a way for destroying sites or making sites. It doesn't by design: Protein interactions inactivate, activate, or occupy sites by changing their state, but they typically don't destroy or construct sites in real time. That is what evolution does. In contrast, in organic Chemistry, the *effective* locus of action—the analog of a site—is a group of atomic agents—a so-called functional group in chemical jargon. Such groups are made and destroyed all the time in chemical reactions. The agents of organic chemistry—the atoms—are situated below the level of effective sites. As an ontological commitment, this is Kappa upside down. Organic chemistry is a radical form of combinatorial construction, whereas protein chemistry is a radical form of combinatorial state change.

A Kappa rule on its own has no biological meaning; it is only the formalization of a factoid, i.e. a decontextualized fact, which has no biological meaning on its own either. Researchers might know how two proteins bind one another, but there is, in general, no clear understanding of why. Answers to why-questions reside at a higher, functional level of organization in which the significance of any given rule is understood in terms of its contribution to the behavior of a system defined by many rules. For example, the binding between two proteins might cause the delay of the propagation of a signal or it might contribute to the amplification of a signal. Rules don't have a causal role on their own. This directs attention to the collective dynamical behavior of rules.

[7] V. Danos and C. Laneve, "Formal molecular biology", *Theoretical Computer Science*, 325/1, 2004, p. 69-110.

[8] P. Boutillier, M. Maasha, X. Li, H. F. Medina-Abarca, J. Krivine, J. Feret, I. Cristescu, A. G. Forbes and W. Fontana, "The Kappa platform for rule-based modeling", *Bioinformatics*, 34/13, 2018, p. i583-i592. V. Danos, J. Feret, W. Fontana, R. Harmer and J. Krivine, "Rule-based modeling of cellular signalling", Concurrency Theory: 18th International Conference, CONCUR 2007. Proceedings, Berlin/Heidelberg, Springer, Lecture Notes in Computer Science Series, vol. 4703, 2007, p. 17-41.

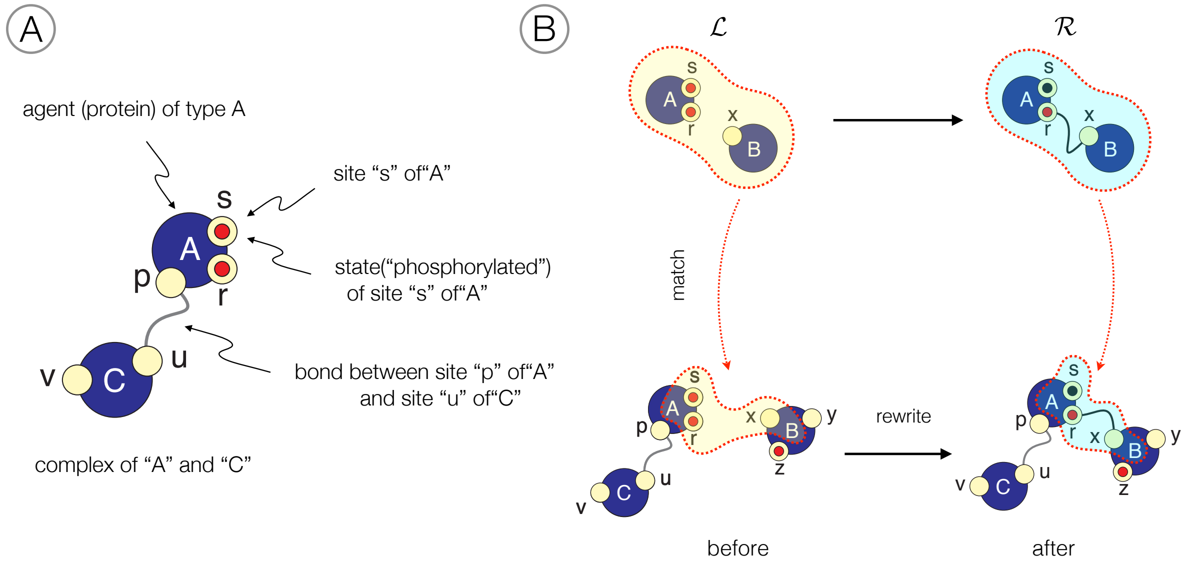


Figure 7: At a level of abstraction often adopted in systems biology, proteins are viewed as agents equipped with sites through which they interact with one another (by binding and post-translational modification) as specified by rules. A complex is a graph composed of two or more proteins linked together at sites. The rewriting rules follow the same formal idea as in organic chemistry.

To generate system dynamics from rules, imagine a virtual protein mixture changing on the basis of repeated and stochastic—that is, probabilistic—applications of rules. At any given moment, the tendency of a rule to apply, and thus to change the state of the mixture, depends on the many ways in which the rule matches its configurations. To visualize rule-based dynamics is challenging, but we might imagine a network whose nodes are rules and whose links depict the influences that the firing of one rule has on the propensity of another to fire. This allows identification of changes that might reflect a restructuring of the causal architecture of the system, for example when some rules stop talking to others. A visualization of this dynamic network of influence is reminiscent of a system of firing neurons.

To properly appreciate the details and behavior of a specific model would require a dedicated lecture on its biological underpinnings and its representation in the Kappa language. Rather, I would like to convey a general sense for the challenge of modeling biology at this mechanistic scale.

One major challenge consists in creating a transparent, computer-assisted process for identifying and translating biochemically and biophysically rich, but often ambiguous, statements from natural language into flat Kappa graphs. Russ Harmer put this succinctly: the challenge is to merge knowledge representation and modeling in such a way that models can become vehicles for storing, tracking, communicating, and analyzing biological knowledge. This presents formidable difficulties that will engage computer science and artificial intelligence. For now, humans merge knowledge representation and modeling in their heads, which is neither scalable nor easily shareable.

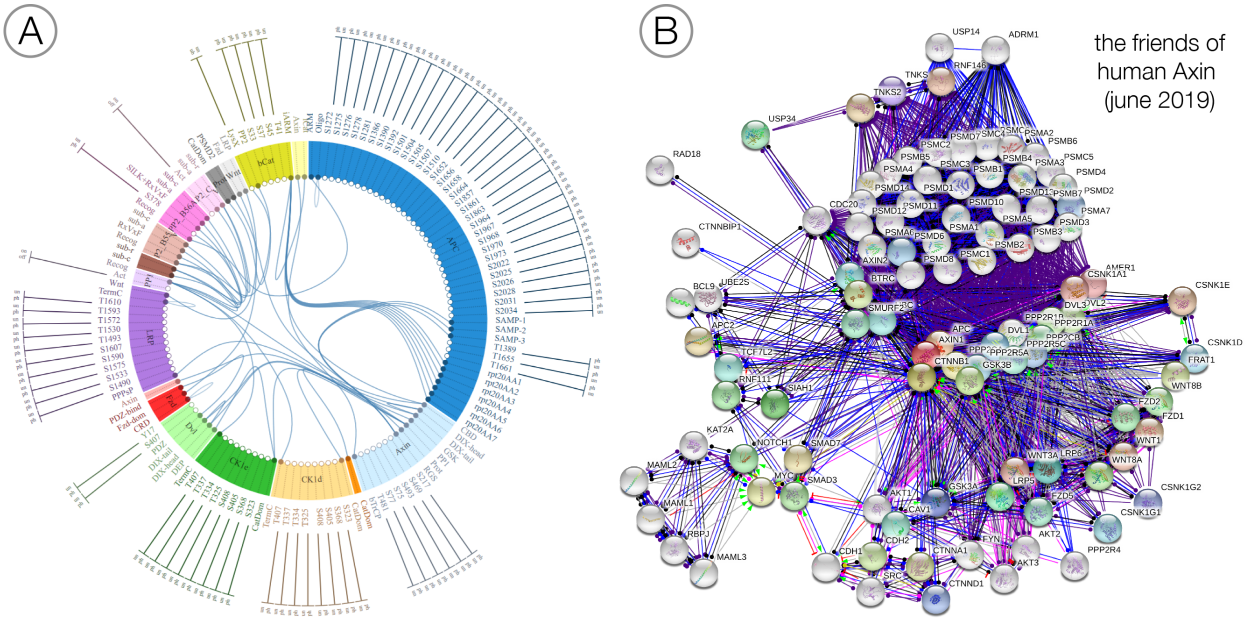
To illustrate further challenges, consider a signaling system in the cell known as the Wnt system[9]. The Wnt signaling sequence, like many other sequences or cascades, plays an important role in embryonic development and cellular maintenance. Its misbehavior is implicated in a variety of

[9] The naming of protein agents in molecular biology consists of acronyms that refer to descriptions of behaviors; a situation that evokes the age of alchemy.

cancers, especially colorectal cancer. When biologists talk about the Wnt system they draw simple diagrams in which arrows connect proteins or protein complexes in a fashion meant to depict a causal progression of events.

The synopsis is roughly as follows: a protein X (β-catenin), in combination with several other proteins, controls the transcription of certain genes in the cell nucleus. In the absence of a signal—the protein called Wnt—X should stay away from the cell nucleus. To prevent X from going there, a complex machinery tags X to be recognized and destroyed by a protein-shredder. When the Wnt signal is intercepted at the cell membrane, a process begins through which the tagging machine is prevented from tagging X, and X can now enter the nucleus.

We would like to replace such a static narrative with a dynamic model that is based on mechanistic facts about how these proteins interact with one another. We translated slightly more than a hundred papers (a vanishing fraction of what is published each year about this system) into Kappa and ended up with a model containing 18 different types of proteins with a total of 57 binding sites, 76 taggable sites at which markers can be placed, 31 rule families describing distinct interaction mechanisms with a total of more than 1300 rules that account for kinetic refinements within those 31 families. Fig. 8A shows a rendering of the components of the model and their possible binding interactions and modification states, not the rules. The 76 taggable sites alone suggest that we are looking at a system with more possible molecular species than there are atoms in the known universe. Because several proteins can also form polymeric structures, the number of possible species is actually infinite. Experimentalists know this, but it does not show up in the typical signaling schemata. It's a representation problem.



the friends of human Axin (june 2019)

Figure 8: (A) An overview of the possible links and states of agents underlying a Kappa model of the Wnt signaling pathway. (B) The empirical network of proteins that can interact with a particular member of the model. Most of them are not even considered in the model. The challenge is to integrate different pathways.

Where should such a model begin and end? What defines its scope? This is not an issue of abstraction level. Kappa defines the abstraction level for us. It also is not a matter of what we know and don’t know, since models can be used to evaluate hypotheses. Rather, it is a problem of entanglement with other signaling systems. Take the protein called Axin. Axin has a “facebook” that lists its friends, which are proteins with which Axin *can* interact. In June 2019, Axin’s facebook looked like Fig. 8B. The facebook also shows interactions among the friends themselves. Each friend of Axin has its own facebook. Even though not all of the friends of Axin, let alone all their friends, are present in the cell at the same time and in the same place, we have no principled idea as to what the boundaries of a model should be. Having a good question certainly helps to focus. Modeling, however, need not always be about answering a question; rule-based modeling in particular can be about finding a good question to ask.

Let us assume that we are satisfied with what our Wnt model covers. Yet, our model comprising 1300 rules is unintelligible, although each rule on its own is perfectly clear. We have replaced a world we don’t understand with a model we don’t understand. We have created a complex artifact that hopefully has some connection to the world of real phenomena. But now we need instrumentation to study it. This is experimental science on a model. It looks like the challenge computer scientists face when they try to understand the complex programs they themselves built.

I want to draw your attention to one particular type of analysis that is particularly suited for rule-based models: causal analysis. The causality I am talking about here is an analysis of what happened in a particular history of events. It is a retrospective causality; the kind of causality that interests the courts in the case of a car accident. The hope is that if we analyze the causality of many histories leading to the same result, we might be able to say something about a more general kind of causality that is predictive, such as “If the brakes of a car are broken, an accident will follow”.

The most detailed output of a model simulation is a long sequence of events in physical time, each one a rule application. With all this information, why is establishing actual causality a challenge? The answer is concurrency. In a probabilistic dynamic, rules behave autonomously. Although we observed a particular history, we could have equally well observed another one, but with *the same outcome*. To understand how a system is organized causally, it is key to classify histories in terms of equivalence classes with each class representing a distinct way of achieving the same goal.

This can be illustrated with the assembly of a desk using a set of rules (Fig. 9). If this were a biological situation, there would be many more rules and hence fundamentally distinct ways of assembling a desk. As an alternative to the case in Fig. 9, one could imagine assembling a desk using a scaffold so one doesn’t need to first mount both side panels; when one gets close to finishing the desk, one disassembles the scaffold. To discover which ways are relevant under which conditions, we let the system itself choose the ways it prefers by simulation. The problem, however, is that trajectories can run in circles. A partially assembled desk can fall apart again only to resume assembly from a state previously visited. A useful causal account is one that contains only steps that were *necessary* to assemble the desk, not simply steps that happened to be on the causal path. This means we must compress away causally futile cycles. After compression, we can reconstruct meaningful causal diagrams. Well-known ideas from computer science allow us to write a single representation for all histories that follow the same type of causal assembly path (Fig. 9). In such a diagram, nodes are rules and an arrow means that the rule at its tail must precede the rule at its tip. We can traverse this diagram by visiting a node in any order as long as we have visited all nodes pointing to it before. This generates all possible equivalent histories that correspond to a particular way of building a desk. If we can characterize all equivalence *classes* of histories, i.e. all ways of

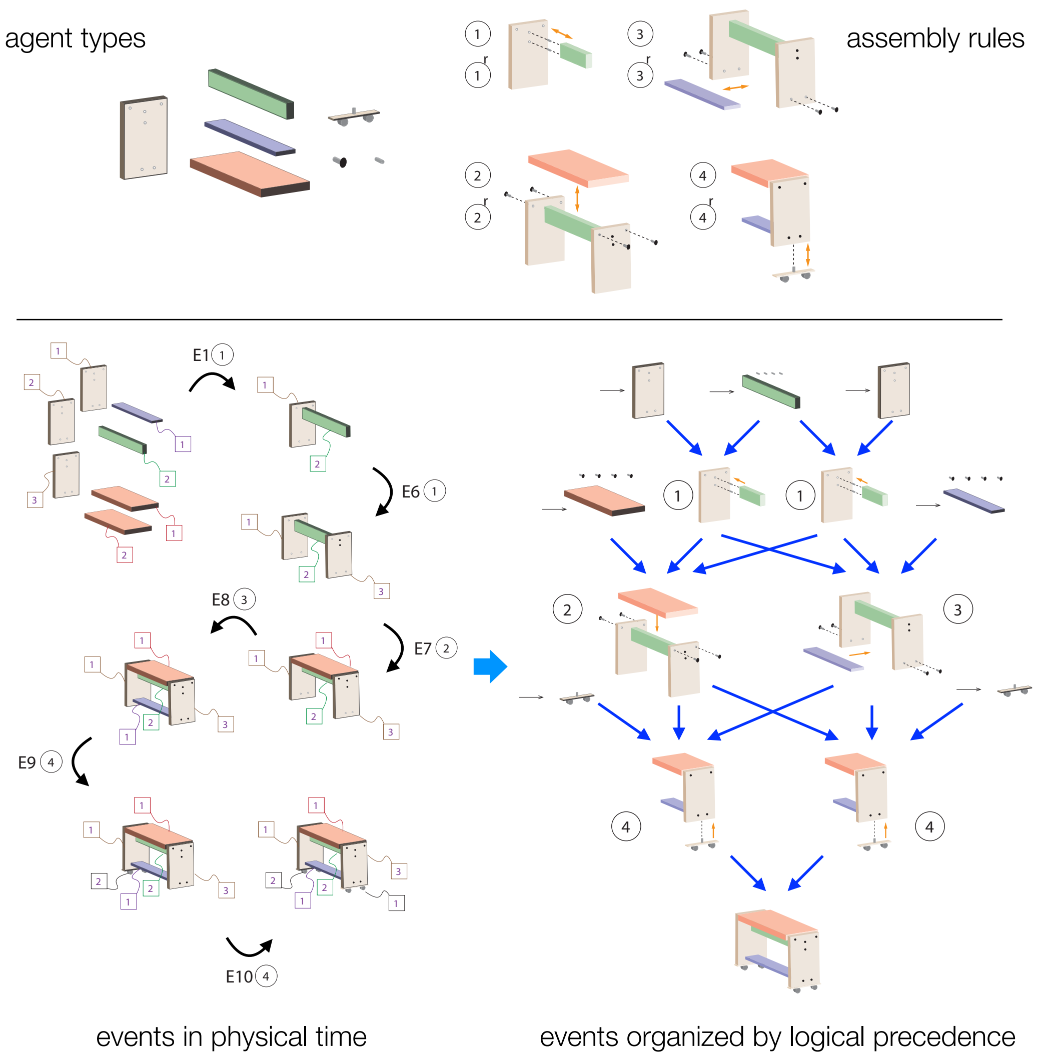


Figure 9: The assembly of a desk. Top: Agent types and assembly rules. Bottom left: A particular assembly history in which cycles have already been compressed away. Bottom right: A diagram showing the causal precedence relation between assembly events. The diagram is derived from analyzing the trajectory on the left. At the same time, the diagram represents a multitude of trajectories that are equivalent to that on the left; they all could have been observed in this particular mode of assembly.

building a desk with a given set of rules, we are a step closer to describing causal organization.

Closure

In closing, I would like to step back and take in the whole picture. I tried to span an arc between three chemistries and representations of their interactions founded on ideas from computer science. When endowed with dynamics, all three give rise to aspects I associate with functional organization.

At the beginning I asked what kind of dynamics *produces* functional organizations, such as the self-maintaining organizations of logic whose change is constrained, or the auto-catalytic chemical networks present in living systems, or the causal structures that organize the signaling processes in cells? This dynamics differs from the dynamics of a particle system that we typically study using differential equations. It is a *constructive* dynamics, which is based on interactions that *directly* build new objects with new interactive properties. It should not be confused with evolutionary dynamics, which is only *indirectly* constructive by acting through the dynamics I just alluded to. The challenge of a science of organization consists in formalizing and understanding this constructive dynamics. It may be difficult, at present, to express this challenge correctly, but I end this lecture trying once more.

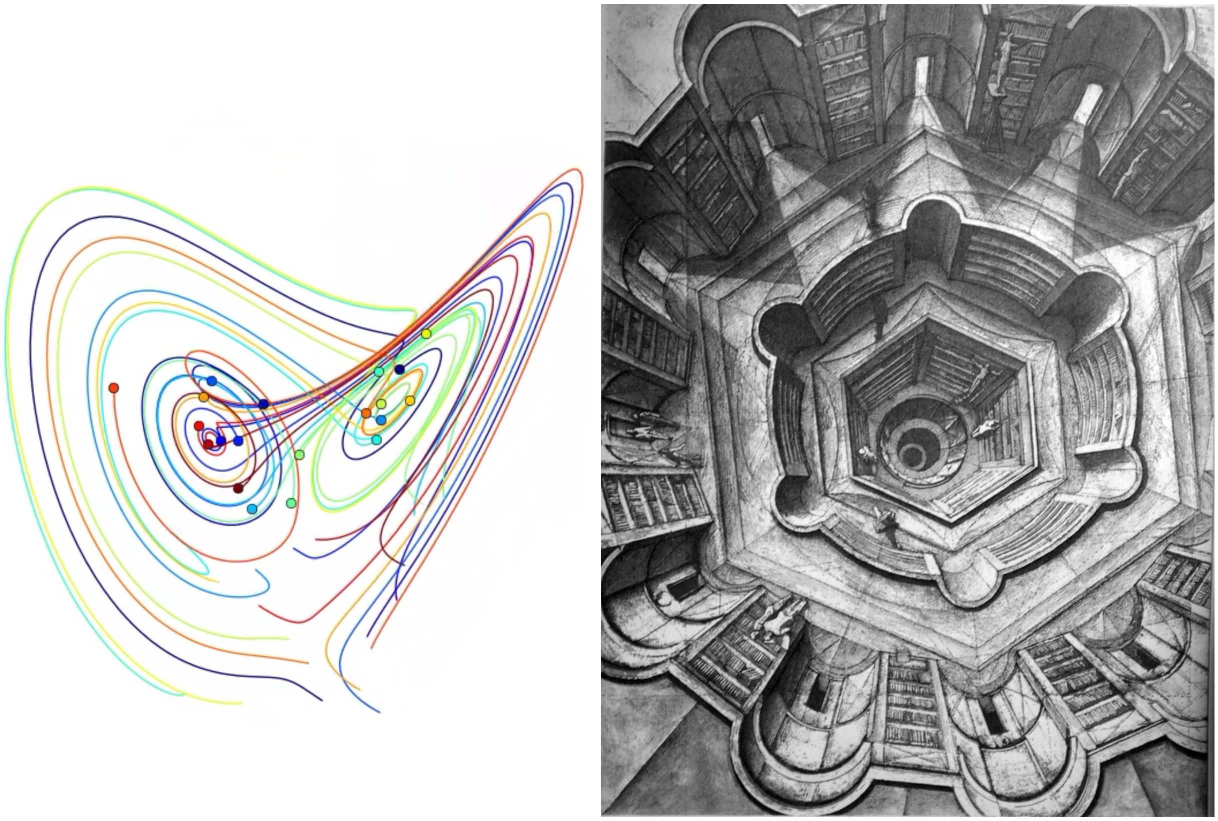


Figure 10: Left: The Lorenz attractor illustrates the idea of a phase space that accommodates the possible trajectories of a dynamical system. Right: An engraving by Erik Desmazières (1997) illustrating "The Library of Babel", a short story by Jorge Luis Borges (1941). In this story, Borges captures the absurdity of a library (a kind of "phase space") that contains all possible combinations of letters of the alphabet (including punctuation) and thus holds, among an infinity of incongruous gibberish, all possible knowledge of the past, present and future. The library is useless, other than driving people mad, because we do not know what to look for. It seems to me an apt metaphor for a chemical space laid out in its totality before reactions have actually built it. Source (engraving): Bibliothèque nationale de France, département des Estampes et de la Photographie, DC-2394-FOL, no 4.

Physics has this wonderfully unifying concept of phase space, as the space of all possible states accessible to a system. It is home to our classical concept of dynamical system. A dynamical system carves tracks in phase space. The dimensions of phase space are linked to the salient dynamical variables of the system, like particle number, momentum, position. For example, there is an axis for momentum, and momentum can take on different values. I have no problem understanding that I have a momentum at this time, even if it is zero. However, I have a problem with the following.

When describing a chemical dynamical system, we typically augment the phase space of physics with one dimension for each of the possible chemical species that can occur in the system. There could be infinitely many of them, but that is not the source of my discomfort. The source sits deeper. The set of chemical species is defined in its totality at the outset alongside all possible chemical reactions in which they participate. This reaction network is then used as a scaffold on which we "hang" chemical kinetics. Chemical kinetics is a dynamics that only changes concentrations. Yet, when we speak of chemistry we really mean processes that can build new molecules, and hence open new dimensions. The construction of new molecules is a dynamics too. However, it is not of the same kind as the dynamics with which we describe, for example, traffic, planetary motion, the growth of a bacterial population or the spread of an epidemic. Rather, it is a dynamics more akin to the ideas of construction that sit at the foundations of computer science. It is a dynamics whose variables are things, not quantities of things. By virtue of generating things, this dynamics also makes possible the quantitative variables associated with them. In the context set by the web of constructive interactions, some of these things will become salient, others will disappear. Yet, in our augmented phase space we have already declared each thing as an available dimension to host a quantity, the concentration of that thing. We have eliminated the dynamics that creates these dimensions. It is as if all the chemistry has already happened so it can host chemical kinetics, which populates the chemical dimensions. There is something fishy about this setup. It is an approach that equates not having (or even knowing) a molecular species with having it as a dimension that is just not yet populated, i.e. at zero concentration. It feels to me like one of those days when I just have no good idea. But instead of saying that I had no good idea today, I proceed with listing all the good ideas I didn't have assigning them a quantity of zero!

One often hears that the challenge of systems biology is quantification. Quantification is not a challenge characteristic of biology; it is a challenge it shares with all sciences. Today, the uniquely difficult challenge of biology is representation. To reason about something you first have to represent it somehow. Computer science will be a fundamental ally of biology because computer science is the science of representation.

I would like to thank Athie Tschibelu for his essential help in the French translation that was delivered in the inaugural lecture. I also gratefully acknowledge Emmanuelle Fleury for her revision of the text.