What is a structural representation?
Fourth variation *

Lev Goldfarb, David Gay, Oleg Golubitsky

Faculty of Computer Science
University of New Brunswick
Fredericton, Canada

October 11, 2005

> [W]e may again recall what Einstein stressed: that given a sufficiently powerful formal assumption, a fertile and comprehensive theory may ... be constructed without prior attention to the detailed facts, or even before they are known.
>
> L. L. Whyte, Internal Factors in Evolution, 1965

Abstract

We outline a formalism for structural, or symbolic, representation, the necessity of which has been acutely felt in all sciences, particularly biology, for quite some time now. At the same time, biology has been gradually edging to the forefront of sciences, although the reasons obviously have nothing to do with its state of formalization or maturity—which is quite primitive as compared, for example, to that of physics. Rather, the reasons have to do with the growing realization that the objects of biology are not only more important and interesting, but that they also more explicitly exhibit the evolving nature of *all* objects in the Universe. It is this view of *objects as evolving structural processes* that we aim to address here, in contrast to the ubiquitous mathematical view of *objects as points in some abstract space*.

One can gain an initial intuitive understanding of the proposed representation by generalizing the (Peano) process of construction of natural numbers: replace the *single* structureless unit out of which a number is built by *multiple* structural ones. An immediate but critical consequence of the distinguishability/multiplicity of units in the construction process is that we can now see *which* unit was attached and *when*. Hence, the resulting representation for the first time embodies temporal structural information in the form of a formative, or generative, history.

* This paper (UNB Faculty of Computer Science Technical Report TR05-174) is a substantially modified variation of [1], [2] (and also [3]). In informal segments of this paper, we often use single quotes to identify standard terms in ETS and other technical areas.

To gain some intuition about the nature of the above “structural unit”, one needs to open it up, i.e. to observe its formation at the *previous level of representation*. To this end, redraw the popular image of particle collision as follows: substitute for each particle track a ‘regular’ structural process (paved with its overlapping ‘generators’, where each generator, in turn, is composed out of structural units from that level), and for the entire collision event, a ‘transformation’ that *restructures* the incoming structural processes into the resulting ones. Thus, in particular, the formalism allows for a very natural introduction of representational levels: a next-level unit corresponds to a previous level transform.

The concept of *class* can then be introduced as that of a class of similar structural processes, where each such process is both a temporal and structural object representation and their similarity is understood via a common generative origin. Furthermore, the proposed concept of *class representation*—which inspired and directed the development of this formalism—differs radically from the few inadequate surrogates that have emerged from numeric formalisms. Indeed, the evolving transformation system (ETS) formalism proposed here is the first one developed to support that concept: a class representation is a generating system that outputs “similar” structural processes, i.e. structured entities serving as object representations (for objects from that class). The process responsible for the construction and modification of both object and class representations is the *inductive learning process*.

The classical discrete “representations” (strings, graphs) now appear as *incomplete* special cases at best, the proper adaptation of which should incorporate corresponding generative histories, as is done here.

The gradual emergence of ETS, including the concepts of structural object and class representations, as well as the associated inductive learning processes and the representational levels, points to the beginning of a new field—*inductive informatics*—which is intended as a *class oriented* rival to conventional information processing paradigms.

Part I
Prolegomenon

> Those who still wish to build a computational empire on the basis of such troubling precedents [i.e. to assume that the Church-Turing thesis is also an adequate scientific and/or epistemological basis for AI] would do well to pause first on the significance of Wittgenstein's ever-timely warning that "One keeps forgetting to go right down to the foundations. One doesn't put the question marks *deep* enough down."
>
> S. Shankar, Wittgenstein's Remarks on the Foundations of AI, 1998

1 Introduction

1.1 Obstacles toward a formalism for structural representation

In this paper we outline a vision of the concept of structural representation which has been in gestation for almost twenty years.

Although the overwhelming importance of structural, or symbolic, representations in all sciences has become increasingly clear during the second half of the twentieth century, there have hardly been any *systematic* attempts to address this topic at a fundamental level[1]. It is not that difficult to understand the main reasons behind this state of affairs. From a theoretical point of view, it appears there are two very formidable obstacles to be cleared: 1) the choice of the central "intelligent" process, the structure and requirements of which would drive and justify the choice of a particular form of structural representation, and 2) the lack of any *fundamental* mathematical models whose roots are not directly related to numeric models. The order in which these obstacles must be addressed is important: obviously, one must first choose which intelligent process to model before attempting to look for a satisfactory formalism. Unfortunately, the second (and principal) of the above obstacles is usually underestimated or overlooked entirely.

Why has it been overlooked? Because, during mankind's *scientific* history, we have dealt only with numeric models and, during the last century, with their derivatives. The latter should not be surprising if we look carefully at the vast prehistory of science in general, and of mathematics in particular [4], [5]. New mathematical abstractions and overspecializations (with a resulting narrowing of historical perspective) during the second half of the twentieth century have also contributed to such a lack of understanding of the extent to which we depend on numeric models[2]. What has (barely) begun to facilitate this understanding, however, is the emergence of computer science in general, and artificial intelligence and pattern recognition (PR) in particular[3].

[1] This situation is particularly puzzling from the point of view of computer science, in view of the central role played by 'data structures' and 'abstract data types'.

[2] There are, of course, rare exceptions (see [6], for example).

[3] Although several "new" areas *very* closely related to PR—such as machine learning (ML), neural networks (NN), etc.—appeared for political reasons during the last twenty years, we will refer to them collectively by the name of the original area, i.e. pattern recognition, or occasionally as inductive learning.

The relevant *concept* of a representational formalism will be discussed in [7]. Here we simply mention that according to the view expressed therein, we presently have (excluding ETS) only one, albeit *very* primitive, representational formalism, i.e. the ubiquitous numeric formalism. In this sense, it is not surprising that the numeric formalism is, basically, the only scientific currency. We aim to change this situation—a goal absolutely unprecedented in the history of science.

The complete monopoly of numeric models in science[4] suggests that it is unreasonable to expect a transition from numerically-motivated forms of representation, which have a millennia-old tradition behind them, to structural forms of representation to be accomplished in one or several papers. At the same time, one should not try to justify, as is often done in artificial intelligence, practically nonexistent progress in this direction by the long standing difficulties involved.

For an extended discussion of related issues, see [9].

1.2 Recent historical perspective: the need for unification

In this work, we outline a fundamentally new formalism—evolving transformation system (ETS)—which is the culmination of a research program originally directed towards the development of a unified framework for pattern recognition [10]–[17].

In view of the fact that newer, more fashionable "reincarnations" of PR (see footnote 3) have missed what is probably the most important representational development within PR during the 1960s and 1970s, we now touch on this issue (which actually motivated the original development of the ETS framework). Over these two decades, it gradually became clear to a number of leading researchers in PR that the two basic approaches to PR—the classical vector-space-based, or statistical, approach and the syntactic, or structural, approach [18], each possessing the desirable features lacking in the other—should be unified [19]:

> Thus the controversy between geometric and structural approaches for problem of pattern recognition seems to me historically inevitable, but temporary. There are problems to which the geometric approach is . . . suited. Also there are some well known problems which, though solvable by the geometric method, are more easily solvable by the structural approach. But any difficult problems require a combination of these approaches, and methods are gradually crystallizing to combining them; the structural approach is the means of construction of a convenient space; the geometric is the partitioning in it.

Although these original expectations for an impending unification were quite high, it turned out that such hopes were quite naive, not so much with respect to timeliness but with respect to the (underestimated) novelty of such a unified formalism: there was no formal framework which could naturally accommodate such a unification [10]. It is interesting to note that researchers working in the various above "reincarnations" of PR have only relatively recently become aware of the need for, and of the difficulties associated with, such an effort. The large number of conferences, workshops, and sessions devoted to 'hybrid' approaches (e.g. [20]–[27]) attests to the rediscovery of the need for unification.

[4] For an insightful explanation of how the stage was set for this, see [8].

1.3 The general direction we have taken

Returning to the two formidable obstacles mentioned in section 1.1, for us and many others the choice of the central intelligent/information process reduced to the pattern recognition process, or more accurately the pattern (or inductive) learning process[5], with an emphasis on the *inductive class representation*. On the other hand, overcoming the second obstacle, i.e. the development of an appropriate mathematical formalism for modeling inductive processes, has been and will be a major undertaking.

What are some of the main difficulties *we* have encountered? In a roughly historical order, they are as follows. On which foundation should the unification of the above two basic approaches to PR be approached? How do we formalize the concept of inductive class representation? How should the Chomsky concept of generativity be revised? How do we generalize the Peano axiomatic construction of natural numbers to the construction of structural entities? In other words, how do we formally capture the more general inductive (or generative) process of object construction? How do we approach the concept of object representation as that of its formative process? What is the connection between a class description, or representation, and the process that generates class objects? How is an object representation connected to its class representation, and, moreover, how do these object representations change during the learning process? How do we introduce representational levels? It is understood that all of the above must be accomplished *naturally* within a single general formalism.

On the formal side, we chose the path of a far-reaching generalization of the Peano axiomatic construction of natural numbers ([32] or [33]), the axiomatics that forms the very foundation of the present construction of mathematics. This choice appears to be a very natural way to proceed. As well-known nineteenth-century German mathematician L. Kronecker aptly remarked, "God made the integers; all the rest is the work of man". Thus, in part, the *original* logic behind the *formalization* was this: take the only existing "representational" model, natural numbers, and generalize the process of their construction/generation, i.e. replace the *single*, essentially structureless primitive out of which natural numbers are built (Fig. 10 (a), p. 26) by *various* structural ones (Fig. 7, p. 22). Then, one can build on that foundation (see also [9]).

1.4 ETS as the first representational formalism

From an ETS perspective, there exist two related concepts of object: the Universe's and an agent's. The Universe's concept of object is that encapsulating the entire *formative history* of that object (as a process in the Universe), while the agent's concept is that associated with the agent's *representation* of the object's formative history (relying, of necessity, on the agent's representational resources). Although we expect both concepts to be captured with the same formal means in the ETS formalism as it is presented here, we will usually pursue

[5] Inductive learning processes have been suggested as being the central intelligent processes by a number of great philosophers and psychologists over the last several centuries (see, for example, [28], [29], [30]). An example of a more recent testament is: "This study gives an account of thinking and judgment in which ... everything is reduced to pattern recognition. ... That pattern recognition is central to thinking is a familiar idea" [31].

the agent's view, unless stated otherwise; in general, the context should clarify which sense is intended.

We now firmly believe that the concept of structural object representation cannot be divorced from that of generative object representation, i.e. that of a representation capturing a generative history of the object. Herein, we believe, lies the fundamental difference between classical numeric and structural, or symbolic, representations. In light of this, widely-used nonnumeric "structural representations" such as strings, trees, and graphs cannot be considered as such. *In short*, it turns out that, since such "representations" do not encode the generative object history[6], there is very little connection between the object representation and the corresponding class representation. Thus, for example, the framework of formal grammars proposed by Chomsky in the 1950s for generating syntactically-correct sentences in a natural language does not address these concerns, which is not quite surprising in view of his repeatedly-articulated opinion about the essential irrelevance of the inductive learning process to cognitive science (see for example [34], [35]). In particular, the generative issues so important to Chomsky simply cannot be properly addressed within the string setting, mainly because a string does not capture the object's formative history; as a result, there are exponentially many formative histories[7] hidden behind a string representation. From the vantage point of the ETS formalism, it becomes clear that the main reason why the various generative grammar frameworks have not succeeded as representational models has to do with their neglect of *more fundamental*, representational issues, i.e. the basic inadequacy of the string as a form of structural representation (see also [9]).

With respect to generative representation, it is useful to note that a number of philosophers and scientists have pointed out the importance of an object's past for that object's representation. Here is a recent example of one such expression [36]:

> [W]e shall argue that memory is always some *physical* object, in the present—a physical object that some observer *interprets as holding information about the past.*
>
> . . . .
>
> . . . The past, about which the object is holding information, is the past *of the object itself.* In fact, an object becomes memory for an observer when the observer examines certain features of the object and *explains* how those features were *caused.*
>
> We shall argue . . . that all cognitive activity proceeds via the recovery of the past from objects in the present. Cognitive activity *of any type* is, on close examination, the determination of the past.

As mentioned in the abstract, we expect that the scientific environment for understanding and investigating the nature of structural representation, together with the concepts of structural object and class representations and its various application areas—e.g. pattern recognition, data mining, information retrieval, bioinformatics, molecular phylogenetics, cheminformatics—will delineate a new information processing paradigm: inductive informatics. To understand the power of this paradigm, it is sufficient to mention how it would

[6] As was mentioned in the previous footnote, the latter should be understood not necessarily in the sense of the *actual* generative history, but rather from the point of view of the *recovered* (class) generation history.

[7] I.e. they correspond, roughly speaking, to sequences of transformations responsible for the formation of this string as an element of a particular class of strings.

transform the widely applicable standard setting of information retrieval, relied on, for example, by all current search engines. Inductive informatics would reduce all such problems to those of 'retrieving' all class elements based either on a small class sample (possibly a single element) or directly on the class representation. Such a uniform formulation becomes possible only with the emergence of the concept of class representation.

In light of the obvious monumental difficulties related to the development of a formalism for structural representation, the best we can hope for as a result of the present attempt is to propose and outline a possible skeleton of such a formalism. We intend to use the proposed outline as a *guide* which will be modified in the course of extensive experimental work in numerous application areas. At the same time, as is always true in science, in our immediate experimental and theoretical work, we will also be guided by a reasonable interpretation of the present tentative formalism (now in its fourth version)[8]. In general, it is important to understand that, when facing such a radical shift in representational formalism, one has *no other choice* but to begin with a theoretical framework, and only then move to the 'data'. Einstein emphasized this point in physics, but in this case the point should be even more apparent, since the notion of *data without a framework for data representation* is absolutely meaningless: it is the framework that dictates how 'data' is to be obtained and interpreted.

Ultimately, what should make or break ETS as a representational formalism? Since it is the first framework explicitly postulating fundamentally new forms of object and class representation, the utility of these forms, as is the case in all natural sciences, can now be experimentally verified. It is interesting to observe that the latter is not possible for any of the current inductive learning models, since they neither insist on, nor even propose, any—formally or otherwise—*meaningful form of inductive class representation*, but simply *adapt* existing formalisms to fit the learning problem (without the availability of adequate concepts of both object and class representations in such formalisms). Thus, one of the immediate values of the ETS formalism is that it is the first formalism developed *specifically* to address the needs of the inductive learning process, and this paper should be interpreted as a *program for action* rather than simply a philosophical deliberation (see also Section 10).

The framework's basic tenets both elucidate the nature of information (or "intelligent") processes in the Universe and are subject to experimental verification. In this respect, it is critical to keep in mind the accumulated scientific wisdom regarding the main value of a scientific model: "Apart from prediction and control the main purpose of science is ... explanation ..." [37] and "Whatever else science is used for, it is explanation that remains its central aim" [38]. *Current inductive learning models explain essentially nothing about the nature of these information processes*, since, as we firmly believe, hardly anything *can* be explained outside an adequate representational formalism.

Finally, ETS suggests a very different picture of reality than that implied by modern mathematics: equational descriptions of physical reality are replaced by structural descriptions (or representations) of evolving classes of objects in the universe. We believe that the proposed formalism provides radically new insight into the nature of objects and classes and offers a guiding metaphor badly needed by various sciences. As to progress in the develop-

[8] Currently, our group is working on applications in such areas as computer vision, speech recognition, bioinformatics, the representation of folk tales, and modeling the game of Go. Unfortunately, due to our very limited resources, the completion and publication of these applications is being prolonged.

ment and applications of ETS, we strongly believe that it would be accelerated within multidisciplinary groups (in which natural sciences are well represented), which, sadly enough, are presently lacking.

1.5 Organization of the paper

The paper is divided into four parts. Part I includes two introductory sections, the second of which outlines a way to think about the 'universe generating process'[9], and Part II (sections 3, 4) presents the basic formal concepts. In Part III (sections 5–8) we introduce the core concepts. Part IV (sections 9, 10) sketches some of our preliminary thoughts on learning and suggests how one should approach the ETS formalism.

In view of the tentative nature of the proposed formalism, it does not make sense to strive for a *very* rigorous form of exposition, and we have followed that wisdom. To streamline the current outline of the formalism, we put our efforts into definitions. As it turns out, even without theorems, the size of the paper is still substantial. Moreover, the predominance of definitions over formal results is easily explained by the absolute novelty of the formalism (including its basic concepts).

Our greatest regret is the lack of at least one working example in this paper. However, the main reason for this absence has to do with our unwillingness to present a non-informative example, i.e. an example that is not *fully* consistent with the underlying assumptions of the framework.

For earlier expositions of the ETS formalism, see [3], [1], [2], and [40]. Some very preliminary applications of the earlier variations of ETS to cheminformatics are discussed in [41], to information retrieval in [42], [43], [44], to bioinformatics in [45], and to speech representation in [46]. Three theses presenting some earlier pre-formal work on the ETS learning algorithms are [47], [48], and [49].

We wish to thank Alexander Gutkin for his extensive comments on various drafts of this paper, Dmitri Korkin for his detailed comments on the present variation, and Thore Graepel for the questions that lead to the addition of two remarks on pp. 22 and 43.

2 The emerging informational view of the Universe

What is the informational structure of the Universe that allows for the development of biological information processing? Above all, a biological information processing model must be fundamentally consistent with a non-biological information processing model: the former was built *on top* of the latter. What we have assumed is that the common central theme (for both kinds of models) has to do with the concept of an evolving class of immediately related objects, where the "description" of the class is closely related to the formative, or generative, structure of its elements.

Before discussing the state of affairs in relation to the concept of class, we present a simple (but, we believe, important) argument supporting the *primacy and uniformity of the*

[9] This term will be immediately clarified in subsection 2.2.

informational view of the Universe, as suggested by ETS and in sharp contrast to conventional scientific paradigms. It is not difficult to see that accepting a qualitative difference between the informational capabilities of the prebiological Universe and of biological species inevitably leads to the acceptance of a scientific principle similar to the well-known and now almost unanimously rejected principle of vitalism[10]. Indeed, if biological species "invented" *fundamentally* new forms of representation that have not previously existed in the Universe, then this is tantamount to saying that biological information processing—and therefore biological mechanisms themselves—cannot be understood on the basis of physical mechanisms. In other words, postulating two fundamentally different informational mechanisms in the Universe *leads to the same undesirable situation* (in view of the ramifications) as postulating two fundamentally different classes of forces acting in the Universe.

2.1 The lack of any adequate concept of class

It is well known that the concept of a class of objects is absolutely pervasive, both within science (e.g. isotope families, biological taxons, categories in cognitive science) as well as outside it (e.g. library classification schemes, fall shoes versus summer shoes). In view of the ubiquity of the class concept, many areas of information processing—e.g. PR (including speech and image recognition), ML, NN, data mining, information retrieval, bioinformatics, cheminformatics—rely on this concept as the central one. So, since the main burden of addressing the concept of class and the process of classification fell on these areas, of necessity they had to settle on some formalisms, and unfortunately, but not surprisingly, the researcher's (subconscious) choice has typically been the classical numeric and logical formalisms. However, as was mentioned in the third last paragraph of section 1.4, these conventional formalisms were not developed to address the needs of class representation, so the researchers have tried to adapt them for this purpose[11], to the extent that, presently, it is these adaptations that have become an obstacle to be overcome. Thus, although—together with most of PR, ML, NN researchers—we firmly believe in the indisputability of the need for a theory of inductive learning, where we differ with others is in our insistence on evaluating such a theory based on the quality of class representation the theory offers: we need a (radically) new formalism that clarifies what the concept of class *is*, since conventional theories contribute practically nothing towards this goal.

In light of this, on the technical side, the development of the ETS model was motivated by two considerations: the fundamental inadequacies of existing formalisms for class description and by the vision of the class description as a 'generating system' (see Section 6). As a result, it turned out that the differences between the conventional (numerical and logical) views of class and class description and those of the ETS formalism are as substantial as they could possibly be: one main difference with numeric formalisms being our insistence on a particular form of generativity in the definition of a class representation (i.e. the class definition must

[10] Vitalism suggests that, in addition to the known physical forces, there are not-yet discovered 'vital forces' that are active in living organisms. Thus, according to vitalists, life cannot be explained by physics alone.

[11] Note that, even in natural languages, words simply *name* classes of objects/events rather than capture their (evolving) *representations*.

also allow one to construct all class elements). Thus, from the ETS perspective, this implies that many widely accepted *verbal* class descriptions, including those often offered in problems in various "ML challenges", should not be considered as such.

2.2 The ETS tenet: evolution of the universe as the evolution of the class generating processes

In general, the ETS framework is inspired by the view of the universe as a variety of interconnected and interacting, evolving generating processes. What do we mean by a generating process?

For a particular class of entities, by its generating process[12] we understand a nondeterministic process operating on structured actual entities and assembling them into larger entities, guided by some abstract description of the class. The latter does not mean that the process "reads" this description, but rather that its instantiation is guided in this particular manner. The appearance of a new generating process is a result of the interaction of several current processes.

As will become clear from our Class Representation Postulate (Section 6), we think of a generating process as a hierarchical generating system—in some sense similar to an abstraction of the embryo's development system[13]—that should be viewed as a class generating system which produces (representations of) class objects. It is important to emphasize that the concept of a generating process, or *class representation system*, now becomes in a sense more fundamental than that of the class itself, i.e. class objects become a product of the corresponding generating process.

Thus, we think of each object, as well as that object's representation, as a *regular* structural process that is being produced, or constructed, by the class generating process. The adjective 'regular', to which we will return in the next subsection, refers to the fact that the same structural 'generators' appear regularly *and* in the same structural setting. For example, a single (vibrating) water molecule, observed over a very short time interval, can be thought of as a regular molecular process, which itself is composed of faster-running subatomic regular processes. Also, the production of *different* water molecules is guided by the *same* generating process, and thus quite naturally, we get a *class* of water molecules. In science, the generating processes have, so far, remained behind the scene, while their products, i.e. class elements, are more familiar to us.

In view of its motivation and structure, we also expect ETS to be the first formalism suitable for modeling developmental processes, a need that been acutely felt within the field of developmental biology:

> *Morphogenesis* remains one of the most poorly understood aspects of development. Although the genetic blueprints underlying the formation of many organs and structures are beginning to be worked out, the mechanisms by which encoded genetic information is translated into structure remain obscure. [50, p. 81]

[12] For a more formal description, see Section 6.
[13] For a popular (but inherently sloppy for a formally-trained reader) exposition, see any of [50], [51], [52], [53].

> [N]o unified theory of modularity that captures the various uses of the term in evolutionary and developmental biology currently exists. *Nor do we have all the formal tools for the analysis of such systems.* [50, p. 341, emphasis ours]

2.3 Regular structural processes and transformations

It has been known for a long time that "[t]ime has its origin *in the existence of both kinds of physical change, cyclic and non-cyclic, in the natural world that we know*" [54, p. 14]. What is the importance of the *two different kinds* of processes?

> Any physical process that is purely cyclical may be treated as an atemporal process as long as it is considered only in its own domain, without reference to the larger, temporal universe surrounding it. The successive oscillations of a photon as it moves through a vacuum, for example, are identical, and there is no change that gives a time coordinate. The motion of an electron about a nucleus, or the cycles of two isolated astronomical bodies rotating about each other, have a similar atemporal quality. On interaction with other systems these purely cyclic processes may become non-cyclic and thereby gain a temporal character.
>
> The cyclic process, then, can become a temporal process only by reference to non-cyclic processes. We may refer to the latter as *progressively* changing processes, in contrast to the cyclic processes in which we do not have progressive change except within any one cycle. We have seen that the progressively changing process does not give us our time concept because change in general does not provide a basis for time standardization and measurement. But likewise, uniform cycles of change will not by themselves give us our time concept, because without progressive change there is no distinction of any one cycle from another. [54, p. 14]

Consistent with the above subdivision of all processes into two categories, in the *current* version of the ETS formalism, the two central concepts (besides, of course, that of class) are those of the *structural process* and the *transformation*, the former corresponding to a generalization of the above "cyclic" process and the latter corresponding to the above "non-cyclic" process.

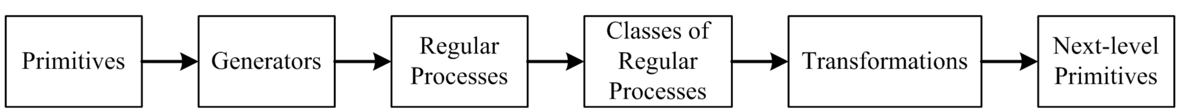

Figure 1: Logical sequence of the central ETS concepts.

Before discussing some of the central concepts, Figure 1 gives their logical sequence. The concept of regular structural process is supposed to encapsulate that of a structural (temporal) object representation—from both the Universe's and agent's perspectives—or, more accurately in the case of the agent's perspective, of an object *observed* over some period of time. From the Universe's perspective, the adjective 'regular' refers to the regular (or periodic) nature of the process responsible for the object's regeneration. From the agent's perspective, 'regular' refers to a simple fact that during the continuous observation of an object, the same 'perceptual features'—or generators in ETS terminology—must reappear regularly as the observation "comes back to the same place". Since *objects are replaced by processes* in the ETS formalism, to more accurately paraphrase the last sentence, the object representation, i.e. the corresponding structural process, is actually constructed during the interaction between the observing (sensing)[14] agent and the target process/object. It is not difficult to see that the choice of generators—out of which a structural process is built—must depend on the agent's (inductive) experience[15].

At a particular level of representation, a generator (at that level) is a temporal assembly of the very basic building units: in our terminology, primitive transformations, or simply primitives. *At the same representational level*, these primitives are treated as indecomposable units designating the basic events associated with the disruption (transformation) of some of the above (indecomposable at this level) cyclic processes (see Fig. 15a on p. 32, where the cyclic processes are shown as lines connecting the primitives). At the *previous* representational level, a primitive can be "opened up" into a *non-primitive* transformation. The nature of such a transformation can most naturally be understood as denoting a macro-event that is responsible for the transformation of one set of "adjacent" regular processes into another set of adjacent processes, i.e. nearby 'initial' processes are transformed into 'terminal' processes [16] (see Fig. 20 on p. 45). These adjacent structural processes are, in fact, opened up prototypes of the next-level indecomposable processes which connect primitives. It is useful to think of a primitive as capturing the (non-regular) event corresponding to the "standard", i.e. regularly reoccurring, interaction of input processes and as also being responsible for the generation of potentially *new classes* of regular structural processes.

In relation to the connection between the above concept of regular process and the concept of the elementary particle in modern physics, the following insight should be useful:

> De Broglie's argument began with the supposition that "the basic idea of quantum theory is the impossibility of considering an isolated fragment of energy without assigning a certain frequency to it." The particles of radiation—and of matter as well—had a *level of existence that was fundamentally a "periodic [i.e. regular] process"*. [55, emphasis ours]

Moreover, as also mentioned on p. 22, the very useful in quantum physics concept of Feynman diagram ([58]–[61]) can be considered as a primitive version of the structural representation proposed here. However, most interestingly, the ETS formalism suggests a very compelling explanation of what some refer to as the most profound mystery in modern physics, i.e. that of entanglement. Entanglement is a term introduced by Schrödinger, and it presently refers (in the case of two particles) to the relatively well observed phenomenon of the *instantaneous* transfer of the effects of measurement on one particle to another particle that interacted with the first one at some earlier time, *independent of the present distance between them* (for

[14] The sensors involved, in contrast to the conventional ones, are structural. It is also important to note that all biological sensors are of a chemical nature, and are therefore structural.

[15] A substantial part of the previous experience of the agent's "species", for efficiency considerations, could be embedded in hardware rather than in software. From a biological point of view, the entire cell structure (including DNA, cytoskeleton, etc.) constitutes such hardware.

[16] Recall the analogy of several particles before and after a collision.

popular expositions, see [62], [63]). From the point of view of the ETS formalism—whose main underlying assumption is the indispensibility of formative history for representation—the above "transfer" of the effects of measurement can be explained as reflecting the following fact: conventional measurements on the first particle contribute to the global formative history (by recording in it the effects of the measurement process itself), which in turn affects all *future* measurement processes (even those addressing "past" events), since they now have to deal with this modified formative history.

In Figs. 2 and 3, we offer an illustration of a naïve ETS representation in physical chemistry, i.e. we present a structural representation of the formation of a lithium hydride molecule from its two constituent atoms (hydrogen and lithium). Note that, for simplicity, otherwise important electron-electron interactions are neglected, nuclei are treated as indivisible or single processes (different for different elements), and the energy, momentum, etcetera of the various subatomic constituents are also neglected. In particular, it is interesting that one does not need to differentiate between atoms and molecules in the ETS formalism[17]: both are regular structural processes at the same level.

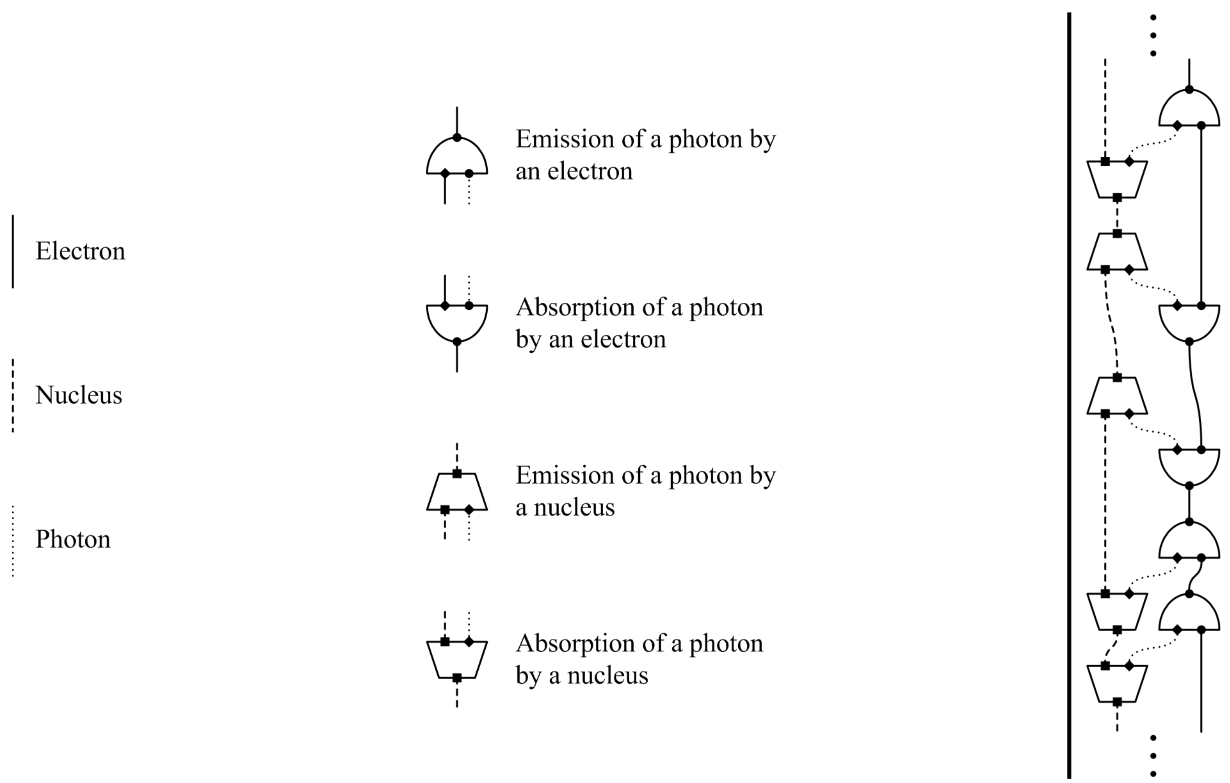

Figure 2: Pictorial representation of a regular process corresponding to a hydrogen atom over an extremely short time interval: three kinds/classes of subatomic processes (first column), four primitive events (second column), and one conceivable event scenario for the hydrogen process (third column).

Turning to a discussion of time scales (see Fig. 4)—and keeping in mind that, in ETS, a transition to a new level is related to the incorporation of a new transformation as a next-level

[17] This does not at all preclude the emergence, at higher levels of representation, of macromolecular structures such as proteins.

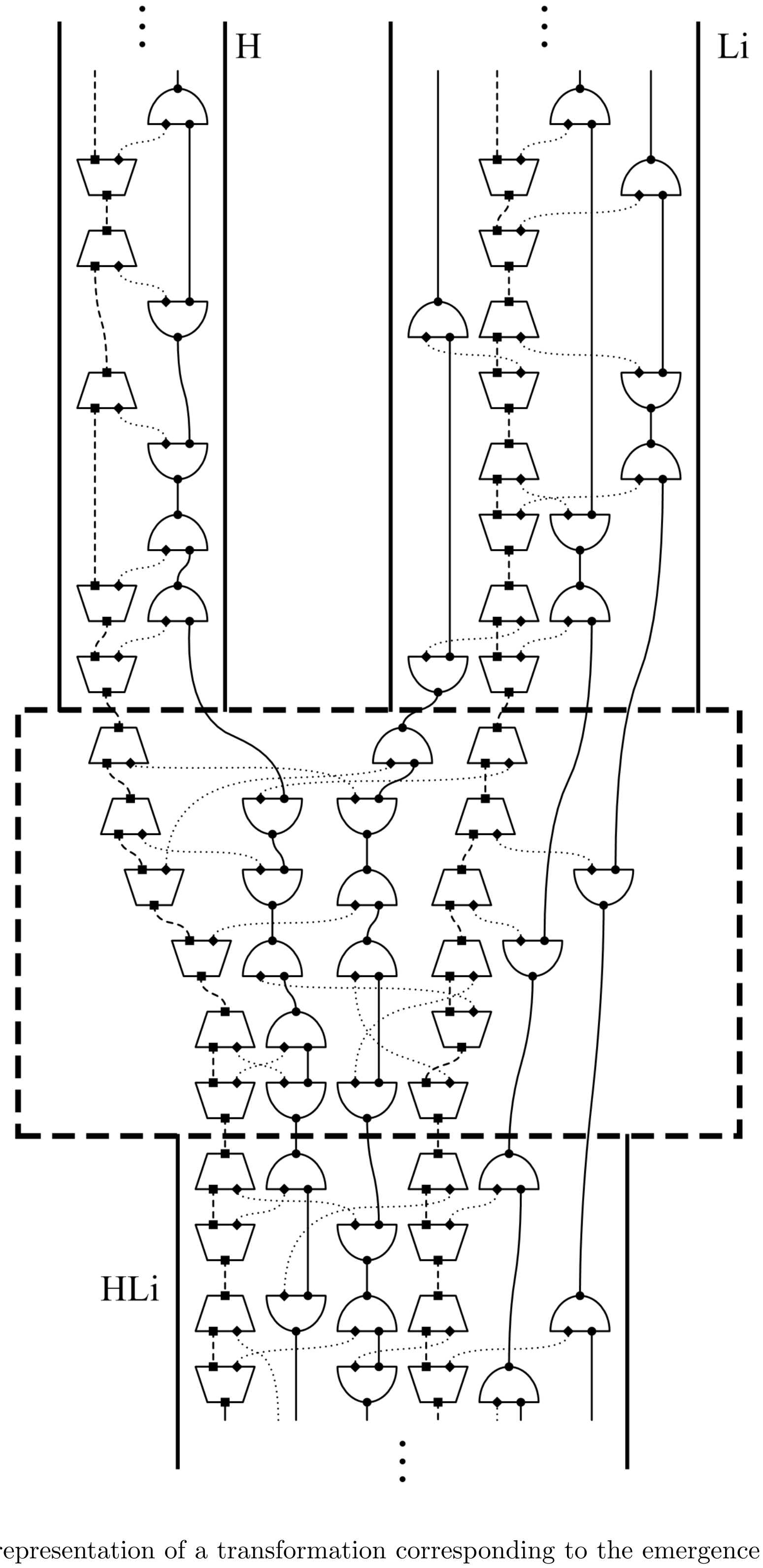

Figure 3: Pictorial representation of a transformation corresponding to the emergence of a lithium hydride molecule (bottom process) from particular hydrogen (Fig. 2) and lithium regular processes (top: note the reocurring regular temporal/structural patterns). The body of the transform (heavy dashed line) captures the restructuring of the two initial processes into the terminal one.

primitive transformation—the ETS formalism suggests that a change in the discretely structured scale of time in the universe is associated with some of the above transitions: a coarser time scale appears as larger (next-level) processes are instantiated. In other words, once the generating process begins to assemble new, larger structural entities, the overall generation time increases. Historically, some of these transitions were associated with transitions to new levels: e.g. atomic levels, molecular levels, etc.

2.4 Objects as epiphenomena of class generating processes

As the section heading implies, according to the ETS formalism, visible objects are not what they appear to be. This point is not as controversial as it seems if all objects are treated as organisms, having developmental as well as evolutionary histories. Thus, a developed organism should also be viewed as an epiphenomenon of *both* these histories as is indeed the case in modern biology: if we tinker with either one of the histories, we change the organism, and the more we tinker, the bigger the changes become. This is what actually happens during evolution. As far as objects are concerned, when we look at such an object as a chair, it also has its "developmental" (i.e. production) and "evolutionary" (i.e. conceptual) histories. In light of this, it makes perfect sense to assume that any biological representational model should be based on such principles: we believe that during the period of time in which a visual system interacts with a chair, for instance, it actually constructs a generative representation of the chair *as perceived by the viewer*. The ETS formalism suggests such a view of reality.

It appears that one can "blame" classical physics for the current state of affairs in which objects (together with the corresponding measurements), rather than their formative histories, are at the center of attention. The latter, in turn, is the result of the state of affairs in mathematics which, historically, has been concerned only with numeric forms of representation. The concept of structural representation proposed here, on the other hand, brings to the fore the question of how the object's structure has emerged. Any non-trivial structure must have emerged incrementally and level-wise, which is what we observe in the universe. For example, molecular structures are now inconceivable without reference to atomic structures. We (together with other scientists and philosophers, e.g. Sec. 1.4) believe that it is such currently unobservable processes that are responsible for the generation of observable structure, and therefore *they* should be of primary interest, as opposed to the observable structures themselves. The latter point of view is quite consistent with the one that emerged about 80 years ago in physics, as can be seen from the following quotation taken from the chapter characteristically titled "Events" in [56], authored by outstanding physicist and astronomer Sir James Jeans (emphasis ours):

> Thus the "world-line" of a particle is, strictly speaking, not a line at all, but is a continuous and unbounded curved region, and must logically be separated into small curved spots—the particle resolves itself into events. *Most of these events are unobservable; it is only when two particles meet or come near to one another that we have an observable event which can affect our senses. We have no knowledge of the existence of the particle between times, so that observation only warrants us in regarding its existence as a succession of isolated events.*
>
> ....

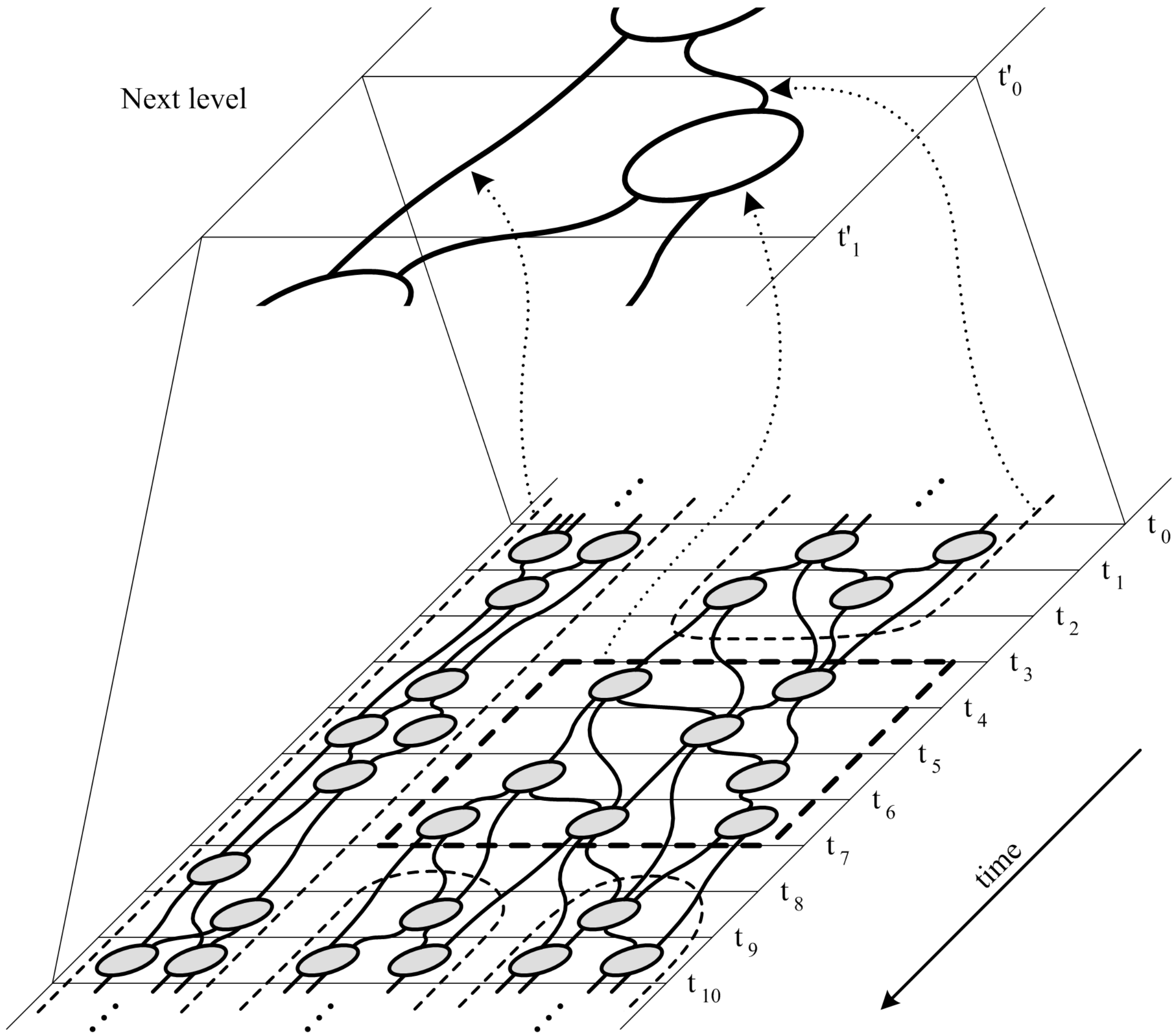

Figure 4: Simplified two-level ETS representation with different time scales for each level. The circles stand for primitive transformations and the lines between them signify basic regular structural processes for the corresponding level. The fine dashed lines identify regular processes and the heavy dashed line identifies the only shown transformation. (Note that the upper level's time scale is measured in coarser units, i.e. t'_0 corresponds to t_0, t'_1 corresponds to t_{10}, etc.)

> Matter gives us a rough and easily understood, but not a true, picture of the reality underlying physical phenomenon. But we now begin to suspect that events and not particles constitute the true objective reality, so that a piece of matter becomes, in Bertrand Russell's words,
>
> "not a persistent thing with varying states, but a system of inter-related events. The old solidity is gone, and with it the characteristics that, to the materialist, made matter seem more real than fleeting thoughts."
>
> This at once takes all force out of the popular objection that mind and matter are so unlike that all interaction is impossible. With matter replaced by events, the objection is no longer tenable. We see the territory on both sides of the mind-body bridge occupied by events, and as Bertrand Russell says (*Outline of Philosophy*, p. 311):
>
> *"The events that happen in our minds are part of the course of nature, and we do not know that the events which happen elsewhere are of a totally different kind."*

2.5 ETS as a multi-level representational formalism

As was mentioned in Section 2.3, the emergence of each new level corresponds to the discovery and consolidation by an agent of the *first* transformation at the currently highest level. Together with the new level, a new primitive transformation is then created on the basis of that transform as shown in Fig. 4. Discovery of a *subsequent* transform at any, except the last, level simply expands the set of primitive transformations at the next level. Since there are several possible informational aspects of (or perspectives on) the Universe's evolution, e.g. subjective versus objective, we now briefly address only the subjective aspect, i.e. an agent's view.

From the agent's perspective, the (external) input is a sequence of sensory events, captured as primitive transformations of the initial level (see Fig. 5). It is quite reasonable to assume, however, that some agents might be equipped with additional kinds of sensors, capable of directly identifying some generators or even classes of regular processes. A useful metaphor for capturing this multi-level structure is a multi-level "evolving representational tower" which can interact with (external) data *only* at the initial level(s). In the language of the ETS formalism, each level records, or represents, 'data' by means of its own primitives, with which it represents the corresponding structural fragment from the previous level (thus compressing the *representation* of the data). Each level k of this tower is responsible for the detection of regularities, i.e. regular processes and transforms, in the data at the k-th resolution level, relying on the (condensed) data representation passed up from the previous level (Fig. 4).

We note that, although it may appear that the transition to a new level of representation is necessitated only by the need to deal more effectively with the complexity of representation[18], the resulting higher level representations in turn begin to play an *active* organizing role influencing the appropriate lower levels.

[18] This results in the well-recognized phenomenon called 'chunking' ([57]).

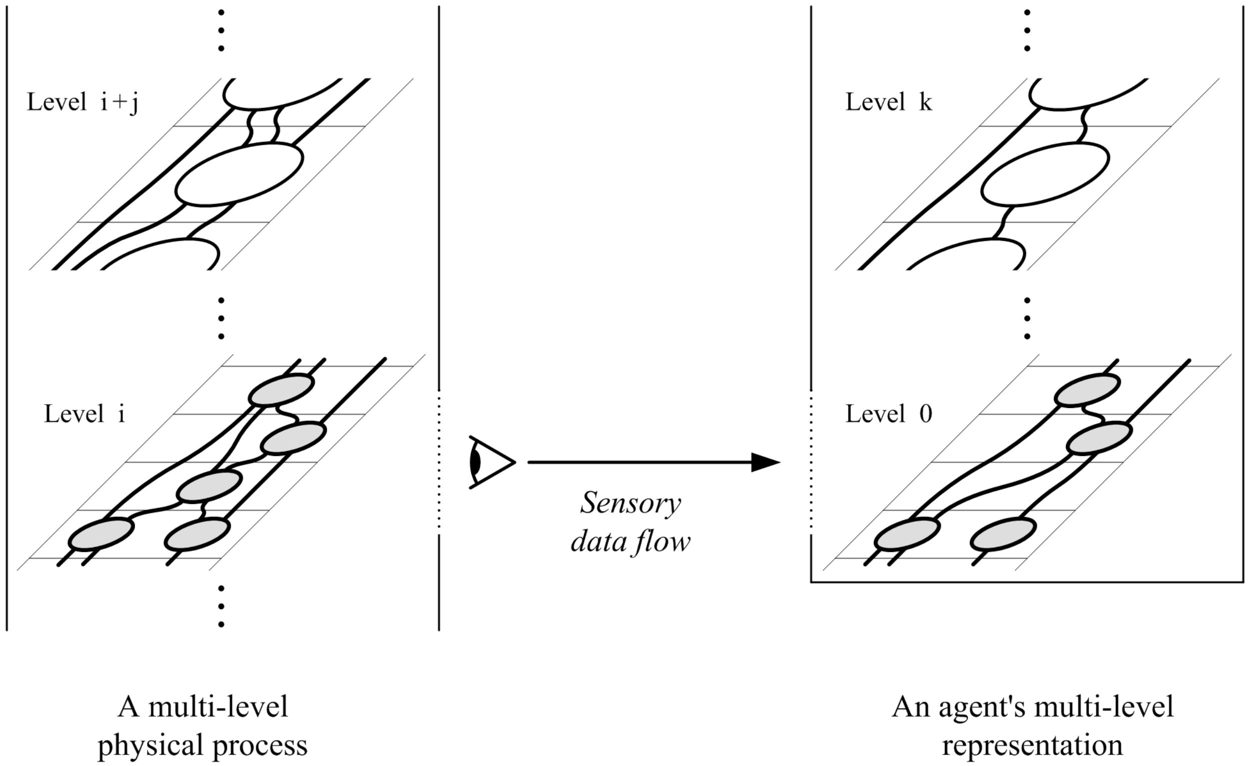

Figure 5: An actual multi-level physical process (left) and an agent's representation of it (right), mediated by sensors (center).

2.6 Coping with the present state of science

In this version of the ETS framework, we propose to cope with the *present* scientific reality with the help of the following two perspectives: *object view* and *event view*. The classical object view encapsulates the *common* scientific view of reality, while the proposed event view encapsulates the ETS, or information process, view.

The conventional scientific view of reality is related to observations in the *object environment*. E.g. in chemistry, observations are those related to atoms and molecules (two separate oxygen atoms covalently bond), and physical theories attempt to describe these observations in terms of states of *objects*. On the other hand, the ETS model emphasizes a *process* view of reality, in which, as was mentioned above, *transforming* events in the object environment, rather than the objects themselves, are the basic subject of study (e.g. in the above example, the focus is on the event corresponding to the *transformation* or *change* responsible for the formation of an oxygen molecule). Again, the latter view of the environment, which we call the *event environment*, insists on the primacy of the information process rather than on the primacy of the objects themselves.

Given the present state of science, i.e. an object-centered view, one may choose to cope with this situation, for the time being, by introducing the above two environments and creating an interface between them: the ETS model operates with **ideal events** that correspond to **real events** in the object environment. A real event is accounted for (in the event environment) by its idealized version (its **idealization**), while in the object environment a real event is accounted for by the **realization** of an ideal event (see Fig. 6). Note that the object environment allows, in particular, such events as the "appearance" and "disappearance" (e.g. in the case of interacting particles) of objects as well as various changes in the relationships among the objects.

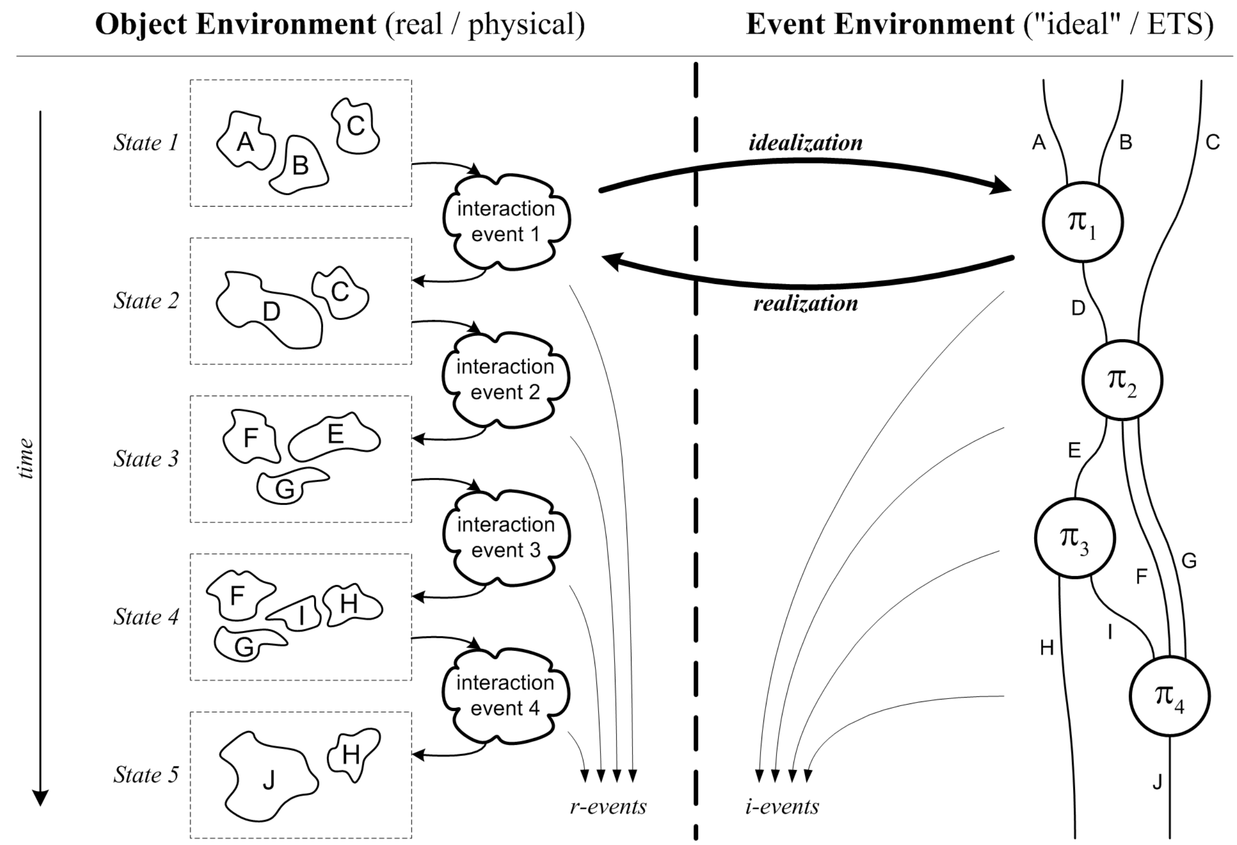

Figure 6: Event environment versus object environment. In State 1, three objects/processes (A, B, C) are shown. As a result of the first real event, A and B merge to form D in State 2. The corresponding ideal event (primitive π_1) is depicted on the right. Three subsequent state changes are also illustrated: D and C are transformed into E, F, and G; E is split into I and H; F, G, and I merge to form J.

Part II
ETS basics

> [T]he above remarks ... prove that whatever the [mathematical language of the central nervous] system is, it cannot fail to differ considerably from what we consciously and explicitly consider as mathematics.
>
> J. von Neumann, The Computer and the Brain, 1958

3 Basic level primitives and class links between them

In this section we introduce the basic constructive elements of the formalism, in our case the elementary transformations (or *elementary 'ideal events'*).

In what follows, the term **enumerable set** refers to either a finite or countably infinite non-empty set.

We would also like to emphasize that, in view of the present lack of an appropriate basic mathematical language for dealing with structured entities, we must of necessity rely on conventional set theoretic language. We do expect that this situation will be remedied once the issue of structural representation has received adequate attention.

Notational convention 1. In this paper, we use the hat accent in $\widehat{\alpha}$ to denote the *special* entity called the name of the associated formal object α. Of course, "naming" can be viewed as an injective mapping from a set of such objects to the set of their names. (From an implementational point of view, this mapping can be realized as a pointer.) ◗

The following definition introduces the first basic concept, that of a primitive transformation. As was mentioned in the Introduction, by 'primitive transformation' we mean a microevent that is responsible for the transformation of one set of "adjacent" structural processes into another set of adjacent processes.[19]

We assume that a set of 'primal' (initial level) disjoint[20] classes of processes has been specified (or chosen). Each element of such a class is some regular *structural process* whose intrinsic structure is suppressed, i.e. each process must, at this initial level, be treated as unstructured, or indivisible, and indistinguishable from any other process in the class. For this reason, in the definition itself we *de-emphasize the concept of 'process'* and speak simply of elements of classes.[21] Still, one *must not forget* that each such element is a regular process (see Remark on p. 43).

[19] Recall the analogy of several particles before and after a collision.

[20] There are reasons to believe that classes (at a *single level* of representation) as they are approached in this paper must be disjoint. Refer to the last paragraph of the *class representation postulate* in Section 6. In general, *assuming complete knowledge of the classes*, including their representations/descriptions, this hypothesis appears to reflect reality.

[21] Clearly, the *real* counterparts to the above abstract processes in the same class must, in *some* sense, be similar to each other. The sense in which they are similar is addressed later, in Section 6.

Definition 1. Let m, n be small positive integers and

$$\mathbf{C} = \{ C_1, C_2, \ldots, C_m \} \qquad C_i \text{ is a given (enumerable) } \textbf{primal class}, \quad \forall\, i,j \quad i \neq j \qquad C_i \cap C_j = \varnothing\,,$$

$$\widehat{\mathbf{\Pi}} = \{ \widehat{\boldsymbol{\pi}}_1, \widehat{\boldsymbol{\pi}}_2, \ldots, \widehat{\boldsymbol{\pi}}_n \} \qquad \widehat{\boldsymbol{\pi}}_i \text{ is a given } \textbf{abstract primitive name}.$$

We first introduce the following auxiliary concepts/notations:[22]

$\mathrm{Init}(\widehat{\boldsymbol{\pi}}_i) = \langle C_{j_1}, C_{j_2}, \ldots, C_{j_{p(i)}} \rangle$ is a given tuple of primal classes called the **tuple of initial classes**, or **initials**, $p(i) > 0$,

$\mathcal{L}_i$, $\mathcal{L}_i \subseteq C_{j_1} \times C_{j_2} \times \ldots \times C_{j_{p(i)}}$, is the **set of labels associated with** $\widehat{\boldsymbol{\pi}}_i$, such that no two constituent elements of any tuple $\langle c_{j_1}, \ldots, c_{j_{p(i)}} \rangle \in \mathcal{L}_i$ are equal,

$\mathrm{Term}(\widehat{\boldsymbol{\pi}}_i) = \langle C_{k_1}, C_{k_2}, \ldots, C_{k_{q(i)}} \rangle$ is a given tuple of primal classes called the **tuple of terminal classes**, or **terminals**.

On the basis of the above givens, define $\boldsymbol{\pi}_i$ as a set

$$\boldsymbol{\pi}_i = \{ \boldsymbol{\pi}_i(a) \mid a \in \mathcal{L}_i \}$$

whose generic element $\boldsymbol{\pi}_i(a)$ is:[23]

$$\forall\, a \in \mathcal{L}_i \qquad \boldsymbol{\pi}_i(a) = \pi_{ia} \stackrel{\text{def}}{=} \langle \widehat{\boldsymbol{\pi}}_i, \mathrm{Init}(\widehat{\boldsymbol{\pi}}_i), \mathrm{Term}(\widehat{\boldsymbol{\pi}}_i), a \rangle\,,$$

see Fig. 7. Set $\boldsymbol{\pi}_i$ is called an **abstract primitive transformation**, or simply **abstract primitive**, and its generic element π_{ia} is called a corresponding (concrete) **primitive**. We denote by $\mathbf{\Pi}$ the finite set of all abstract primitives $\boldsymbol{\pi}_i$, $1 \leq i \leq n$ and by Π the set of all (concrete) primitives.

[One should think of a concrete primitive π_{ia} as designating a particular kind of interaction between the (initial) processes in label-tuple a, the outcome of which is a *non-deterministically*[24] specified element of $C_{k_1} \times C_{k_2} \times \ldots \times C_{k_{q(i)}}$, which is the reason why label a cannot point to a particular element in the latter set product: if one were to observe the event denoted by a single primitive, all of its terminal processes are "in progress".] ▶

As was mentioned in the Introduction, in general, one can think of an abstract primitive $\boldsymbol{\pi}_i$ as an entity responsible for the generation of potentially *new classes* of structural processes. In a formal setting, however, an abstract primitive can be thought of as simply transforming a tuple of initial classes, $\mathrm{Init}(\widehat{\boldsymbol{\pi}}_i)$, into a tuple of terminal classes, $\mathrm{Term}(\widehat{\boldsymbol{\pi}}_i)$.

[22] Note that we do not forbid, for example, that $C_{j_1} = C_{j_2}$. Moreover, the index $p(i)$ of $j_{p(i)}$ simply signifies a natural number that is a function of i.

[23] Both of the following notations, i.e. $\boldsymbol{\pi}_i(a)$, π_{ia}, will be used. Note that π_{ia} should more accurately be interpreted as $[\pi_i]_a$.

[24] This is true since a terminal process cannot be fully identified until it is "absorbed" by another primitive event (see beginning of Section 4).

Figure 7: Pictorial illustration of two abstract primitives (left) and three corresponding concrete primitives (right). The last two concrete primitives belong to the second abstract primitive. The initial classes are marked as various shapes on the top, while the terminal classes are shown on the bottom. (As was mentioned in the definition, the label—a, b, or c, in our case—of a concrete primitive points to a specific tuple of elements from the initial classes.)

Remark 1 (neurons). We would like to draw one's attention to the similarity between our primitives and biological neurons, which in this case is less superficial than that between the units of (artifical) neural networks and neurons: we expect that a neuron plays a similar *representational* role as a primitive does and also that their representational structures are analogous. ◗

Remark 2 (selection of primitives). Regarding the selection of primitives, one cannot overestimate the following point: as was mentioned above, the ETS formalism insists on approaching the representation of 'data' in a much more careful manner than is traditional in information processing and other sciences. ETS suggests that, from an informational point of view, data should now be understood and treated in a *generative setting*. In other words, as will be discussed in Section 6, data emerges already partitioned into classes, where the elements of a class are processes produced by the corresponding class generating system. Thus, a concrete data representation must now be treated as the result of an *appropriate* class generating process, ensuring that the object and class representations are properly correlated. This is definitely not the case with conventional generative (including graph) grammar models, in which the *nature of data representation* is not addressed and is, therefore, not at all correlated with the corresponding generative grammar (e.g. strings do not come with grammars embedded in, or attached to, them). Hence, the selection of primitives should be approached in this light: they are basic events in a carefully chosen process view of the (applied) environment, i.e. *the representation of an object refers to a relevant structural process in such an environment.* In particular, we strongly recommend that all attempts to adapt conventional discrete "representations" (e.g. strings, trees, and graphs) to the above generative setting be abandoned, as they *impede the (inductive) discovery of the relevant generative mechanisms based on the hypothesized object's formative history.* ◗

It is interesting to mention that there exists an immediate analogy between our primitives and the 'vertices' of Feynman diagrams (see, for example, [58]–[61]). However, the approach proposed here is a more careful and general development, from the more abstract point of view of a representational formalism.

Notational convention 2. To simplify the notation in what follows, instead of $\mathrm{Init}(\widehat{\boldsymbol{\pi}})$ or $\mathrm{Term}(\widehat{\boldsymbol{\pi}})$, we will use the notation $\mathrm{Init}(\boldsymbol{\pi})$ or $\mathrm{Term}(\boldsymbol{\pi})$, respectively, i.e. we drop the hat accents.

Also, for the above primitives $\boldsymbol{\pi}_i$ and π_{ia}, we will use the following convenient notations:

$\overline{s}(\boldsymbol{\pi}_i\,, r)$ as referring to class C_{j_r} in the tuple $\mathrm{Init}(\boldsymbol{\pi}_i)$,
$\underline{s}(\boldsymbol{\pi}_i\,, r)$ as referring to class C_{k_r} in the tuple $\mathrm{Term}(\boldsymbol{\pi}_i)$. ◗

The following definition might be viewed as a continuation of (or as closely related to) the previous one: in contrast to Definition 3, it introduces two closely connected relations that are *all potential*—but *not yet observed*—relations among pairs of primitives.

Definition 2. Based on the definition of the abstract primitive, one can introduce the following relation $\mathbf{CL}_{\mathbf{C},\boldsymbol{\Pi}}$,

$$\mathbf{CL}_{\mathbf{C},\boldsymbol{\Pi}} \subseteq \boldsymbol{\Pi} \times N_{\mathrm{Term}} \times \boldsymbol{\Pi} \times N_{\mathrm{Init}} ,$$

with

$$N_{\mathrm{Term}} = \Big\{ u \in \mathbb{N} \;\Big|\; 1 \le u \le \max_i \big(|\mathrm{Term}(\boldsymbol{\pi}_i)| \big) \Big\}$$
$$N_{\mathrm{Init}} = \Big\{ v \in \mathbb{N} \;\Big|\; 1 \le v \le \max_j \big(|\mathrm{Init}(\boldsymbol{\pi}_j)| \big) \Big\},$$

defined as follows

$$\mathbf{CL}_{\mathbf{C},\boldsymbol{\Pi}} = \Big\{ \langle \boldsymbol{\pi}_i\,, u_i\,, \boldsymbol{\pi}_j\,, v_j \rangle \;\Big|\; \boldsymbol{\pi}_i\,, \boldsymbol{\pi}_j \in \boldsymbol{\Pi}, \quad \underline{s}(\boldsymbol{\pi}_i\,, u_i) = \overline{s}(\boldsymbol{\pi}_j\,, v_j), \quad u_i \le |\mathrm{Term}(\boldsymbol{\pi}_i)|, \quad v_j \le |\mathrm{Init}(\boldsymbol{\pi}_j)| \Big\}.$$

We call this relation a **class link between abstract primitives**. The natural numbers u_i and v_j wil be referred to as the u_ith **terminal site of** $\boldsymbol{\pi}_i$ and the v_jth **initial site of** $\boldsymbol{\pi}_j$, respectively.

We also apply similar terminology to the corresponding (concrete) primitives and speak of the (enumerable) relation $\mathrm{CL}_{\mathbf{C},\Pi}$

$$\mathrm{CL}_{\mathbf{C},\Pi} \subseteq \Pi \times N_{\mathrm{Term}} \times \Pi \times N_{\mathrm{Init}} ,$$

defined as follows

$$\mathrm{CL}_{\mathbf{C},\Pi} = \Big\{ \langle \pi_{ia}\,, u_i\,, \pi_{j\mathfrak{b}}\,, v_j \rangle \;\Big|\; \pi_{ia}\,, \pi_{j\mathfrak{b}} \in \Pi, \quad \underline{s}(\boldsymbol{\pi}_i\,, u_i) = \overline{s}(\boldsymbol{\pi}_j\,, v_j), \quad u_i \le |\mathrm{Term}(\pi_{ia})|, \quad v_j \le |\mathrm{Init}(\pi_{j\mathfrak{b}})| \Big\}.$$

Since, *in contrast to the situation with initial processes*, none of the (concrete) terminal processes for *separately considered* concrete primitive π_{ia} can be identified (see the last paragraph in Def. 1), it is natural to postulate that, if the above pair $\langle \pi_{ia}\,, \pi_{j\mathfrak{b}} \rangle$ of concrete primitives has been *actually observed*, the non-determinism with respect to the corresponding concrete terminal process is eliminated, i.e. what has in fact been observed is a concrete process “connecting” the two primitives. In other words, we postulate that the observed “connecting” process is, in fact, the v_j'th element in label $\mathscr{b}$. We will call this relation a **class link between concrete primitives** (see Fig. 8).

For any set Π_0, $\Pi_0 \subseteq \Pi$, we will use the notation $\mathrm{CL}_{\mathbf{C},\Pi_0}$ to denote the subset of $\mathrm{CL}_{\mathbf{C},\Pi}$ in which all π_{ia}'s and π_{jb}'s are from Π_0. ▶

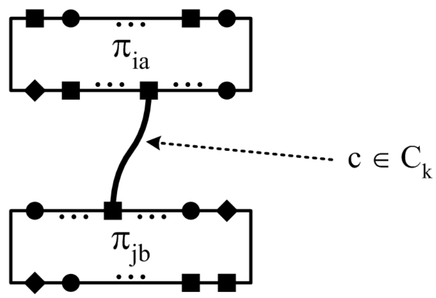

Figure 8: Generic element of the relation $\mathrm{CL}_{\mathbf{C},\Pi}$, where c is a structural process from class C_k.

Thus, a class link between concrete primitives signifies the fact that one of the structural processes produced by the first primitive becomes (at certain point in time) an “input” to the second primitive.

4 Structs and struct unification

From the point of view of an agent interacting with its environment, it is quite natural to expect that, in addition to the *ability to observe primitives*, the agent's sensors must have the *capability of recording the interrelationships* among observed pairs of primitives. (The latter is accomplished by detecting primal structural processes first, and then inferring the primitives which are “connected” by them.) These primitives and their interconnections can be thought of as representing a *macroevent*, or “structural history”.

Definition 3. A **struct** σ is defined as the following pair

$$\sigma \;=\; \langle\, \Pi_\sigma, \mathrm{SL}_\sigma \,\rangle\,,$$

where Π_σ is a finite subset of Π satisfying the condition that, for any two of its elements $\pi_{ia}, \pi_{jb} \in \Pi_\sigma$, no constituent element of tuple $\mathscr{a}$ is equal to a constituent element of tuple $\mathscr{b}$, and the relation **struct link** SL_σ is a finite subset of $\mathrm{CL}_{\mathbf{C},\Pi_\sigma}$ such that:

$\forall \langle\, \pi_{ia}\,,\, u_i\,,\, \pi_{jb}\,,\, v_j \,\rangle \in \mathrm{SL}_\sigma \quad (\, \mathscr{a} \in \mathcal{L}_i,\;\; \mathscr{b} \in \mathcal{L}_j\,)$

(i) the *directed* graph of the following (auxiliary) binary relation[25] representing the projection of SL_σ onto $\Pi_\sigma \times \Pi_\sigma$

$$\mathrm{ATTACH}_\sigma \;=\; \Big\{\, \langle\, \pi_{ia}\,,\, \pi_{jb} \,\rangle \;\Big|\; \langle\, \pi_{ia}\,,\, u_i\,,\, \pi_{jb}\,,\, v_j \,\rangle \in \mathrm{SL}_\sigma \,\Big\}$$

[25] The vertices of the graph are the elements of the relation's underlying set and the edges are defined by the ordered pairs of the relation. We will call this relation **attachment between primitives**.

is acyclic[26]

(ii) $\forall \langle \pi_{ia}, u_i, \pi_{j\mathfrak{b}}, v_j \rangle, \langle \pi_{i'a'}, u_{i'}, \pi_{j'\mathfrak{b}'}, v_{j'} \rangle \in \mathrm{SL}_\sigma$

$$\pi_{ia} = \pi_{i'a'}, \; u_i = u_{i'} \iff \pi_{j\mathfrak{b}} = \pi_{j'\mathfrak{b}'}, \; v_j = v_{j'},$$

i.e. any terminal site can be connected to at most one initial site, and visa versa.

The set of all structs will be denoted Σ, and when $\Pi_\sigma = \varnothing$, struct σ will be called the **null struct**, denoted θ. ▶

Thus, when forming a struct σ, ATTACH_σ is the result of the sensors having recorded the various observed interrelationships among pairs of primitives. Note the role of the ATTACH_σ relation: two *separately-observed* primitives cannot reliably be attached to each other, since, by Def. 1, the specification of a terminal class yields an "incomplete" element of that class; however the "completed" element is obviously necessary to equate this terminal with a particular initial (which is a completed element of an initial class). Hence, the basic role of a sensor is to specify the relation ATTACH_σ, and therefore structs themselves.

We note that *when drawing* a struct σ, its primitives should be positioned in the following way: for any two *attached* primitives, the top primitive should be the one that "precedes" the bottom one in the binary relation ATTACH_σ (see Fig. 9).

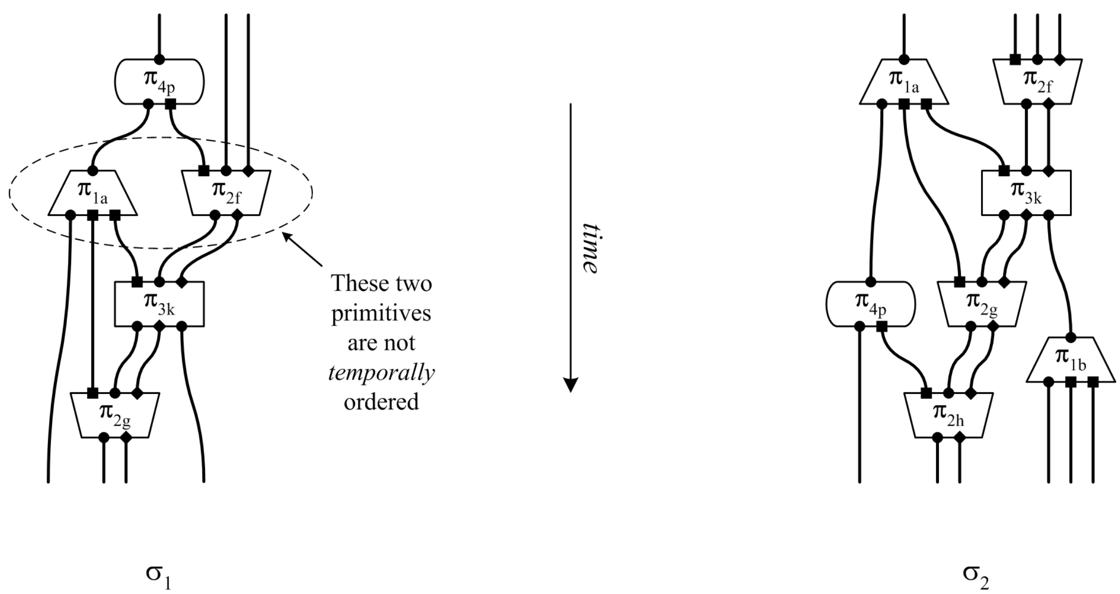

Figure 9: Two structs σ_1 and σ_2.

It is important to note that the above definition suggests a far-reaching structural generalization of the Peano (inductive) construction of natural numbers ([5], [4], see also Fig. 10): we still deal with the temporal, or inductive, order of steps (not structures), in which

[26] A graph without cycles is called acyclic.

small sets of primitive transforms are added atemporally in a *single* inductive step (e.g. $\{\pi_{1a}, \pi_{2f}\}$ in σ_1 in Fig. 9). However, the resulting binary relation between primitives cannot now be interpreted, or understood, as a simple linearly ordered relation but is a more complex structure.

(a) (b)

Figure 10: (a) The single primitive involved in the ETS representation of natural numbers. (b) Structs representing the numbers 1, 2, and 3.

The following is an auxiliary concept.

Definition 4. For a struct σ and two of its primitives π_{kc}, π_{ia}, we will say that π_{kc} is a **descendant** of π_{ia} in σ, if there exists a *directed* path from π_{ia} to π_{kc} in the directed graph of ATTACH_σ.

The set of all descendants of π_{ia} in σ will be denoted $\text{Desc}_\sigma(\pi_{ia})$. ▶

We now introduce a part-whole relation between two structs.

Definition 5. For two structs $\sigma_1 = \langle \Pi_{\sigma_1}, \text{SL}_{\sigma_1} \rangle$ and $\sigma_2 = \langle \Pi_{\sigma_2}, \text{SL}_{\sigma_2} \rangle$, we say that σ_1 is a **substruct** of σ_2, denoted $\sigma_1 \Subset \sigma_2$, if

$$\Pi_{\sigma_1} \subseteq \Pi_{\sigma_2} \qquad \text{and} \qquad \text{SL}_{\sigma_1} \subseteq \text{SL}_{\sigma_2} .$$

▶

The following important but non-central definition will be useful throughout the paper.

Definition 6. By a **relabeling** we understand an injective mapping f with domain $\mathcal{L}$, $\mathcal{L} \subseteq \bigcup_{i=1}^{n} \mathcal{L}_i$ ($\mathcal{L}_i$ is the set of labels associated with $\widehat{\boldsymbol{\pi}}_i$),

$$f \colon \mathcal{L} \to \bigcup_{i=1}^{n} \mathcal{L}_i$$

such that

$$\forall\, i \qquad f(\mathcal{L} \cap \mathcal{L}_i) \subseteq \mathcal{L}_i .$$

▶

To relate two structs based on their common substructures (for example, see Fig. 11), we must be able to "intelligently" (but consistently) modify the incidental labeling of the constituent primitives of one of the structs in such a way that the substructures, i.e. substructs, of interest become identical.

Definition 7. For a struct

$$\sigma \;=\; \langle\, \Pi_\sigma \,,\, \mathrm{SL}_\sigma \,\rangle$$

and a relabeling $f\colon \mathcal{L} \to \bigcup_{i=1}^{n} \mathcal{L}_i$, where

$$\mathcal{L} \;\supseteq\; \{\, a \mid \boldsymbol{\pi}_i(a) \in \Pi_\sigma \,\}\,,$$

the **relabeled struct**

$$\sigma\{f\} \;\stackrel{\text{def}}{=}\; \langle\, \Pi_{\sigma\{f\}} \,,\, \mathrm{SL}_{\sigma\{f\}} \,\rangle$$

is defined as

$$\begin{aligned} \Pi_{\sigma\{f\}} \;&=\; \Big\{ \pi_{if(a)} \;\Big|\; \pi_{ia} \in \Pi_\sigma \Big\} \\ \mathrm{SL}_{\sigma\{f\}} \;&=\; \Big\{ \big\langle\, \boldsymbol{\pi}_i\big(f(a)\big)\,,\, u_i \,,\, \boldsymbol{\pi}_j\big(f(b)\big)\,,\, v_j \,\big\rangle \;\Big|\; \big\langle\, \boldsymbol{\pi}_i\big(a\big)\,,\, u_i \,,\, \boldsymbol{\pi}_j\big(b\big)\,,\, v_j \,\big\rangle \in \mathrm{SL}_\sigma \Big\} \end{aligned}$$

(see Fig. 11). ▶

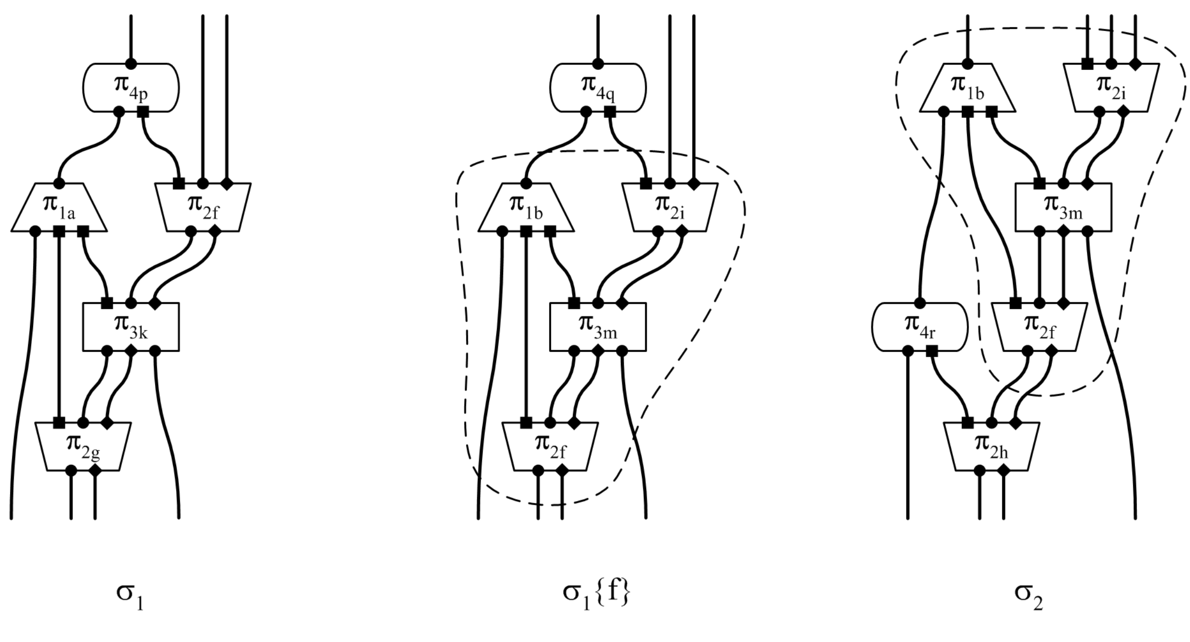

Figure 11: A struct σ_1 (left), a relabeled struct $\sigma_1\{f\}$ (center), and a third struct σ_2 (right). Relabeling reveals some structural similarity (dashed line) between $\sigma_1\{f\}$ and σ_2.

The following definition introduces the concept of a *structurally identical* class of structs, i.e. those structs which differ *only* in the (provisional) labelings of their primitives.

<u>Definition</u> 8. Two structs σ_1 , σ_2 will be called **structurally identical**, denoted $\sigma_1 \sim \sigma_2$, if

$$\exists f \qquad \sigma_2 = \sigma_1\{f\}\,.$$

The corresponding equivalence class containing σ_1 will be denoted $[\![\sigma_1]\!]$. ▶

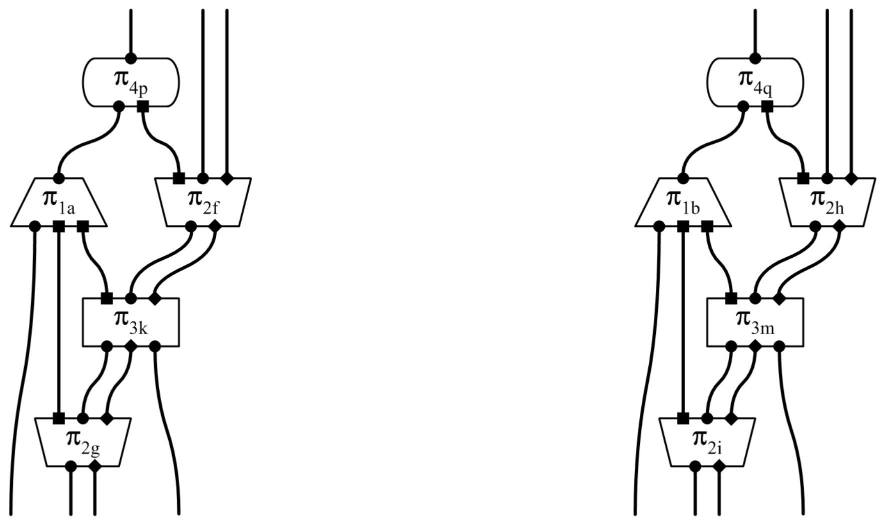

Figure 12: Two structurally identical structs.

We now introduce the basic operation on structs, in which the involved structs are allowed to overlap. The latter condition is useful, for example, when putting together several observed but overlapping structs.

<u>Definition</u> 9. Two structs $\sigma_1 = \langle\, \Pi_{\sigma_1}\,,\, \mathrm{SL}_{\sigma_1} \rangle$ and $\sigma_2 = \langle\, \Pi_{\sigma_2}\,,\, \mathrm{SL}_{\sigma_2} \rangle$ are said to be **unifiable**, denoted $\sigma_1 \curlyvee \sigma_2$, if the pair

$$\langle\, \Pi_{\sigma_1} \cup \Pi_{\sigma_2}\,,\, \mathrm{SL}_{\sigma_1} \cup \mathrm{SL}_{\sigma_2} \rangle$$

is a struct. In this case the above pair defines the struct called the **union** of σ_1 with σ_2, denoted $\sigma_1 \Cup \sigma_2$. It is easy to see that the above relation and operation are associative and commutative, and therefore can be naturally extended to any finite number of structs. ▶

It is easy to see how the concept of 'intersection' of two structs can be introduced: $\sigma_1 \Cap \sigma_2 = \langle\, \Pi_{\sigma_1} \cap \Pi_{\sigma_2}\,,\, \mathrm{SL}_{\sigma_1} \cap \mathrm{SL}_{\sigma_2} \rangle$. The reason this concept does not appear in Def. 9 relates to our desire to focus the reader's attention on the more fundamental operation on structs.

We note that if two structs are not unifiable, it means that they represent an inconsistent, or contradictory, view of reality: it is not possible that both corresponding macroevents occur. Also note that it is quite possible that the resulting (legal) unification of two structs may leave them disjoint if they don't share any concrete primitives, i.e. if they themselves

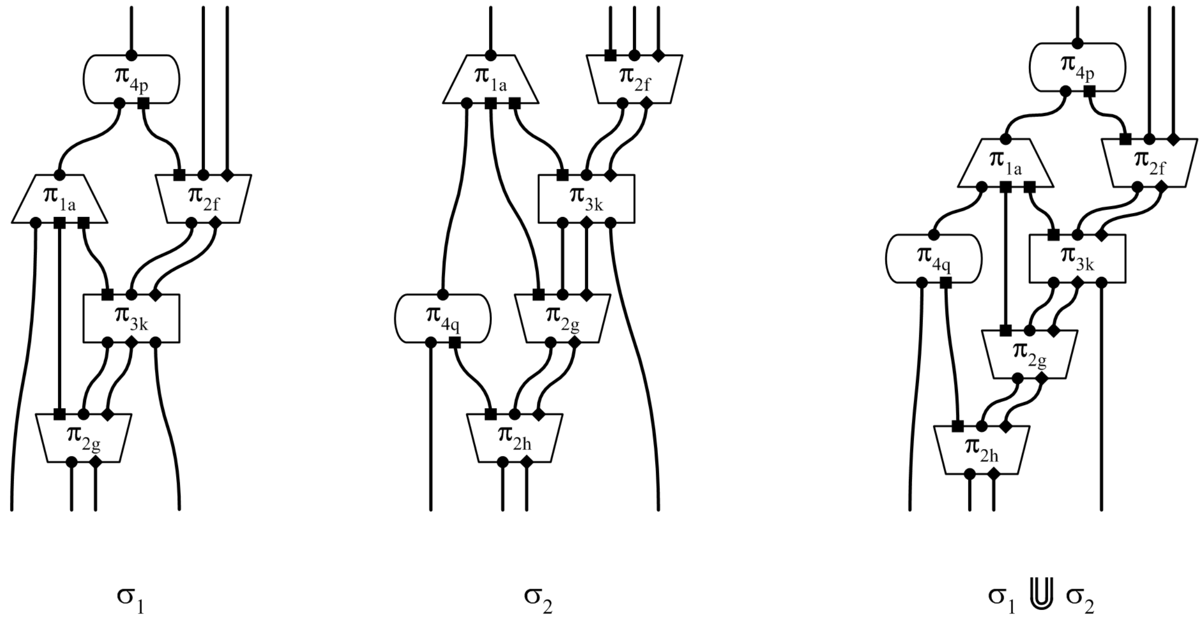

Figure 13: Two structs and their union.

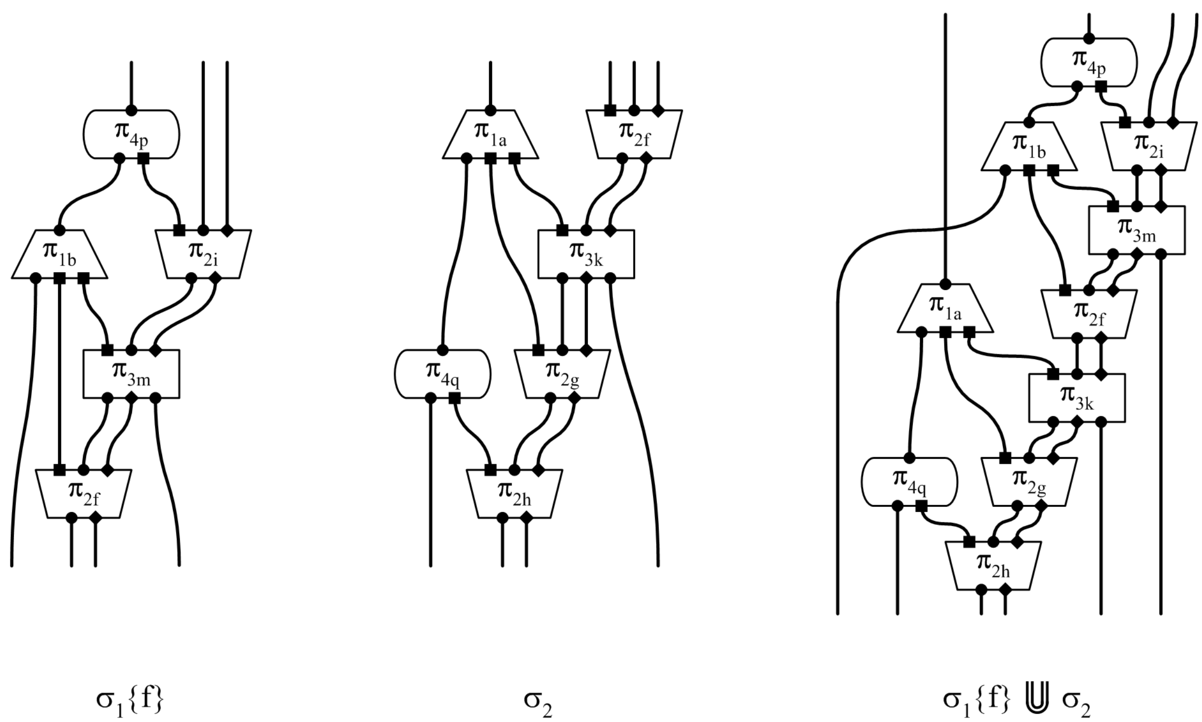

Figure 14: The two structs from Figure 13, the first one of which is *relabeled*, and their union.

are disjoint. Moreover, it is important to note that the legality and result of the unification of two structs are affected by the particular labeling of their primitives, i.e. a relabeling of one of the structs may change the *legality* and/or *result* of their unification (see Figs. 13 and

14).

Observe that, similar to the above situation with attaching two separately-observed primitives (see p. 25), two separately-observed structs *with no overlap* (i.e. with no primitives in common) cannot reliably be "attached" to each other. The only way to resolve this potential ambiguity is to actually *observe* a third struct that overlaps both of the structs in question.

Lemma 1. For two substructs σ_1 and σ_2 of σ, we have

$$\sigma_1 \Subset \sigma\,,\ \sigma_2 \Subset \sigma \qquad \sigma_1 \curlyvee \sigma_2 \quad \Longleftrightarrow \quad \sigma_1 \Cup \sigma_2 \Subset \sigma\,. \qquad \blacksquare$$

It is not difficult to see that the following properties hold:

(i) for a struct σ and relabelings f of struct σ and g of struct $\sigma\{f\}$, we have

$$\big(\sigma\{f\}\big)\{g\} \quad = \quad \sigma\{g \circ f\}$$

(ii) for structs σ and γ, if $\sigma = \gamma\{f\}$, then there exists the **inverse relabeling** f^{-1} of struct σ such that $\sigma\{f^{-1}\} = \gamma$

(iii) if $\alpha \curlyvee \beta$ and f is a relabeling of struct $\alpha \Cup \beta$, then $\alpha\{f\} \curlyvee \beta\{f\}$ and

$$\alpha\{f\} \Cup \beta\{f\} \quad = \quad \big(\alpha \Cup \beta\big)\{f\}\,.$$

Part III
ETS core

> The Universe does not consist of ready, finished objects, but instead represents a collection of processes in which objects continuously appear, change, and are destroyed. Nevertheless, from this it does not follow that they [objects] do not have a definite form of existence, that they are unstable, or that they are indistinguishable among themselves. However much the object changes, up to a certain point, it remains particularly that—and not any other—qualititatively definite object. . . .
>
> *Quality* is the essential definiteness of an object, due to which it is, first, *that* object and not any other, and second, that it is different from other objects. The quality of an object, as a rule, is not reducible to its individual properties; rather it is connected with the object as a whole, captures it fully, and is inseperable from it. [Note that, in ETS, it is the concept of class representation that captures this concept of quality.] . . . Together with qualitative definiteness, all objects also possess quantitive definiteness: definite size, number, volume, ...
>
> *Quantity* is that definiteness of an object due to which (in reality or in thought) it could be subdivided into homogeneous parts [see also Fig. 10] that are then agglomerated. Homogeneity (the resemblance, similarity) of parts of objects is the distinguishing feature of quantity.
>
> *The distinctions between objects that are not similar to each other carry qualitative character [i.e. they belong to different classes], while distinctions among similar objects carry quantititive character.* . . . The exceptionally broad applicability of mathematical theories in the different domains of natural sciences and engineering can be explained by the fact that mathematics studies mainly quantitative relationships. Quality cannot be reduced to quantity, as metaphycisists attempt.
>
> Entry on "Quality and Quantity" in *Philosophical Dictionary*, ed. I. T. Frolov, 5^{th} edition, Moscow, 1987 (our translation from Russian and our emphasis)

5 Generators and regular processes

In this section, we would have preferred to present a more *complete*, or *satisfactory*, view of the central concepts of a regular process and of a class of regular processes. However, what one actually finds here is just a preliminary sketch of those concepts, which is supposed to lead to a radically new view of the Universe as that composed solely of *interacting and evolving regular processes*.

The following concept of a generator is supposed to capture the idea of a small pattern of primitives, which occur regularly in structs and which, together with other such patterns, "pave" structs. Such small patterns of primitives are allowed to overlap each other.

<u>**Definition**</u> **10.** A **process generator**, or simply **generator**, $\mathfrak{g}$ is a pair of structs (Fig. 15)

$$\mathfrak{g} \;=\; \langle\, \alpha\,,\, \gamma \,\rangle\,,$$

where $\alpha \Subset \gamma$, and, introducing the notation

$$\Pi_\beta \;=\; \Pi_\gamma \setminus \Pi_\alpha \,,$$

the following conditions are satisfied:

(i) for each primitive π in α, all of its *farthest descendants*[27] are in Π_β

(ii) $\forall\, \pi \in \Pi_\beta$, $\mathrm{Desc}_\gamma(\pi) \subseteq \Pi_\beta$.

For the above generator $\mathfrak{g}$, we define its **body**, $\mathrm{body}(\mathfrak{g})$ as the following struct

$$\mathrm{body}(\mathfrak{g}) \;=\; \beta \;=\; \langle\, \Pi_\beta \,,\, \mathrm{SL}_\beta \,\rangle$$

where

$$\mathrm{SL}_\beta \;=\; \Big(\mathrm{SL}_\gamma \setminus \mathrm{SL}_\alpha\Big) \setminus \Big\{ \langle\, \pi_{ia},\, u_i,\, \pi_{j\mathfrak{b}},\, v_j \,\rangle \;\Big|\; \pi_{ia} \in \Pi_\alpha \,,\, \pi_{j\mathfrak{b}} \in \Pi_\beta \Big\} \,.$$

Finally, the above struct α will be called the **context** of $\mathfrak{g}$, denoted $\mathrm{cont}(\mathfrak{g})$ (see Fig. 15a). ▶

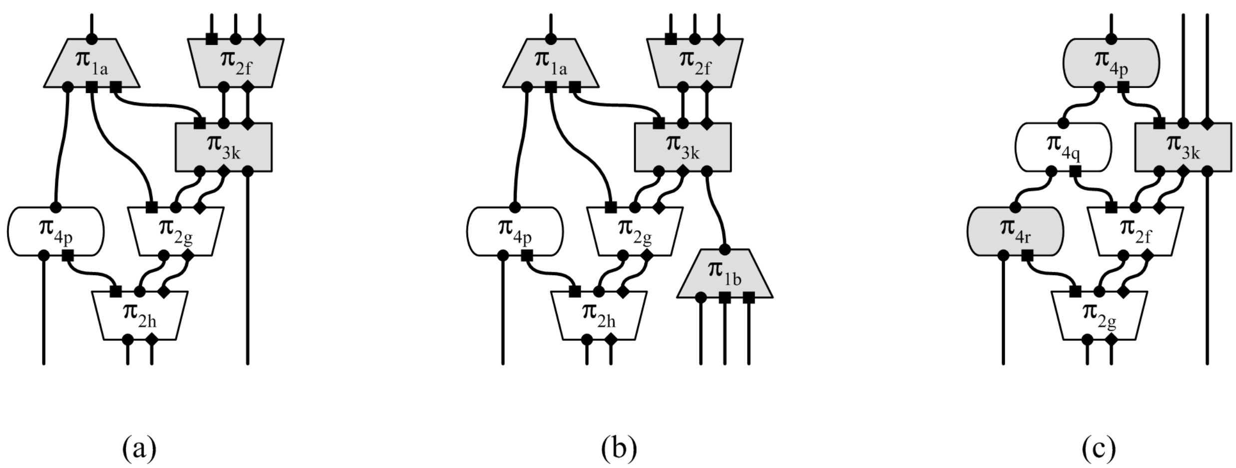

Figure 15: (a) A generator $\mathfrak{g} = \langle\, \alpha\,,\, \gamma \,\rangle$ with shaded context α. (b) For this struct, condition (i) is violated by $\pi_{1\mathfrak{b}}$. (c) For this struct, condition (ii) is violated by π_{4r}.

Thus, the above generator $\mathfrak{g}$ should be thought of as consisting of a context α and body β, where the context must occur before the body and is a precondition for the appearance of the body, i.e. every primitive in the context must have at least one descendant in the body.

How does a generator $\mathfrak{g}$ "apply" to a struct σ?

[27] A descendant of primitive π in a struct σ is called *farthest* if it has no descendants of its own.

Definition 11. For each generator $\mathfrak{g} = \langle \alpha , \gamma \rangle$, we define the associated operation on the set Σ of structs as the following mapping

$$\triangleleft_{\mathfrak{g}} : \Sigma \to \Sigma$$

$$\forall \sigma \in \Sigma \qquad \triangleleft_{\mathfrak{g}}(\sigma) = \begin{cases} \sigma \Cup \gamma & \text{if } \alpha \Subset \sigma \text{ and } \sigma \curlyvee \gamma \\ \sigma & \text{otherwise.} \end{cases}$$

In the case when $\alpha \Subset \sigma$ and $\sigma \curlyvee \gamma$, the result of the above operation is called a **continuation of σ by generator $\mathfrak{g}$** and is denoted

$$\triangleleft_{\mathfrak{g}}(\sigma) = \sigma \triangleleft \mathfrak{g}$$

(see Figure 16). ▶

Thus, the notation $\sigma \triangleleft \mathfrak{g}$ implies that the *restricted* operation $\triangleleft$ can, in fact, be carried out (i.e. $\alpha \Subset \sigma$ and $\sigma \curlyvee \gamma$). Moreover, when several such operations appear in the specification of a struct, the construction of such a struct is understood as proceeding in a sequential manner, from left to right.

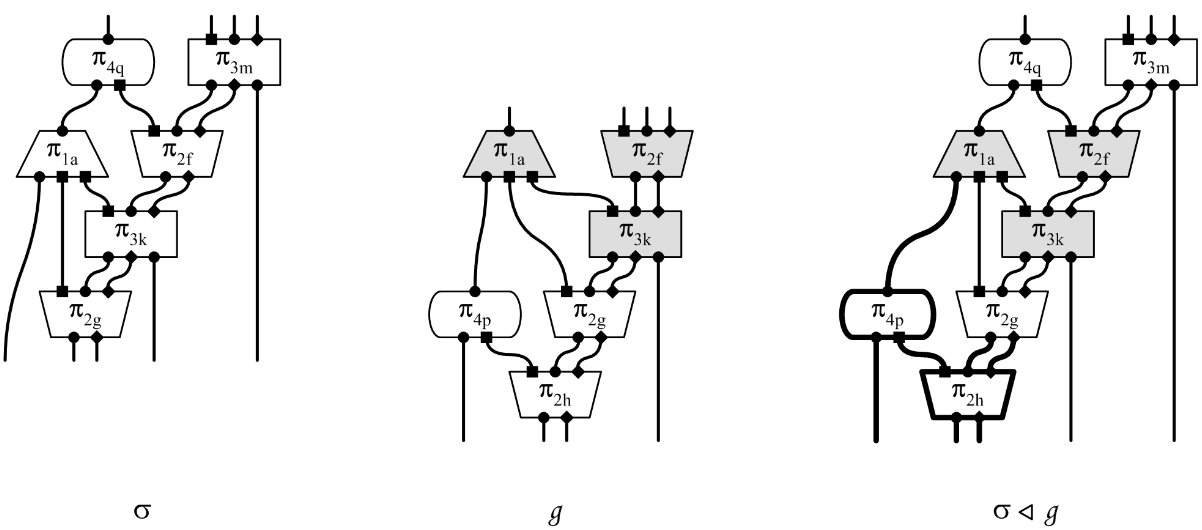

Figure 16: Pictorial illustration of the operation of continuation of σ by generator $\mathfrak{g}$. Primitives in the context are shaded, while new primitives "added" to σ as a result of the continuation are shown in bold.

As was the case with structs, we need to be able to relabel generators and form classes of structurally identical generators.

Definition 12. For a generator $\mathfrak{g} = \langle \alpha , \gamma \rangle$ and a relabeling f of γ, the **relabeled generator** is defined as

$$\mathfrak{g}\{f\} = \langle \alpha\{f\} , \gamma\{f\} \rangle .$$

▶

The next definition is a natural generalization of the concept of structurally identical structs (Def. 8) to that of structurally identical generators.

<u>Definition</u> 13. Two generators $\mathfrak{g}_1$, $\mathfrak{g}_2$ will be called **structurally identical**, denoted $\mathfrak{g}_1 \approx \mathfrak{g}_2$, if

$$\exists f \qquad \mathfrak{g}_2 = \mathfrak{g}_1\{f\}\,.$$

The corresponding equivalence class containing $\mathfrak{g}_1$ will be denoted $[\mathfrak{g}_1]$ and will be called an **abstract generator**. ▶

Having fixed some struct σ and a finite family of abstract generators, we are ready to introduce the following definition of a set of $\langle\sigma, \mathfrak{G}\rangle$-regular processes. Such a process is supposed to characterize a stable, or regular, segment of a structural process, or, more accurately, a struct segment *produced by* an evolving structural process. (A more adequate approach, involving the mechanism behind the regularity, should emerge on the basis of the Class Representation Postulate, introduced in the next section.) Under such an over-simplified view, each stable segment of a structural process is constructed as a particular series of continuations of struct σ with various generators.

<u>Definition</u> 14. Let σ be a struct, and let $\mathfrak{G}$ be a finite family of abstract generators

$$\mathfrak{G} \;=\; \big\{\, [\mathfrak{g}_1],\, [\mathfrak{g}_2],\, \ldots,\, [\mathfrak{g}_r]\,\big\}\,,$$

where

$$\mathfrak{g}_i \;=\; \langle\, \alpha_i\,,\, \gamma_i \,\rangle\,.$$

First, for a fixed $l \in \mathbb{Z}_+$, select such an l-tuple

$$\langle\, \mathfrak{g}^\star_{j_1}\,,\, \mathfrak{g}^\star_{j_2}\,,\, \ldots\,,\, \mathfrak{g}^\star_{j_l} \,\rangle \qquad 1 \le j_n \le r,\;\; \mathfrak{g}^\star_{j_n} \in [\mathfrak{g}_i],\;\; \mathfrak{g}^\star_{j_n} = \langle\, \alpha^\star_{j_n}\,,\, \gamma^\star_{j_n} \,\rangle\,,$$

$$(\,[\mathfrak{g}^\star_{j_n}] \text{ could be equal to } [\mathfrak{g}^\star_{j_m}]\,,\text{ for } n \neq m)$$

that the following struct construction can be carried out (Def. 11)

$$\sigma \lhd \mathfrak{g}^\star_{j_1} \lhd \mathfrak{g}^\star_{j_2} \lhd \ldots \lhd \mathfrak{g}^\star_{j_l}\,;$$

then, we can construct the following struct $\gamma^\star$

$$\gamma^\star \;=\; \gamma^\star_{j_1} \Cup \gamma^\star_{j_2} \Cup \ldots \Cup \gamma^\star_{j_l}\,.$$

Define the enumerable set of structs

$$\Gamma_\sigma(\mathfrak{G}) \;=\; \{\, \gamma^\star \in \Sigma \mid \gamma^\star \;=\; \gamma^\star_{j_1} \Cup \gamma^\star_{j_2} \Cup \ldots \Cup \gamma^\star_{j_l} \quad \text{for all possible above } l\text{-tuples}\,\}\,,$$

called a **set of regular σ-continuation structs associated with $\mathfrak{G}$**, or simply **$\langle\sigma, \mathfrak{G}\rangle$-regular structs**.

For the above $\langle\sigma,\mathfrak{G}\rangle$-regular struct $\gamma^\star \in \Gamma_\sigma(\mathfrak{G})$, the above l-tuple $\langle\, \mathfrak{g}^\star_{j_1}\,,\, \mathfrak{g}^\star_{j_2}\,,\, \dots\,,\, \mathfrak{g}^\star_{j_l}\,\rangle$ is denoted $\mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma^\star)$ and is called a **$\langle\sigma,\mathfrak{G}\rangle$-representation**, or simply **generator representation, of struct** $\gamma^\star$.

Finally, define the enumerable **set of regular σ-continuation processes associated with $\mathfrak{G}$**, or simply **$\langle\sigma,\mathfrak{G}\rangle$-regular processes**, as the set of pairs

$$\mathrm{RP}_\sigma(\mathfrak{G}) \;=\; \Big\{ \Big\langle\, \gamma^\star\,,\, \mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma^\star)\, \Big\rangle \;\Big|\; \gamma^\star \in \Gamma_\sigma(\mathfrak{G}) \Big\}$$

(see Figs. 17, 18).

The set of all regular processes, i.e. for all σ and $\mathfrak{G}$ is denoted RP. ▶

Lemma 2.

$$\Big\{ \gamma \in \Sigma \;\Big|\; \Big\langle\, \gamma\,,\, \mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma)\, \Big\rangle \in \mathcal{R} \Big\} \;=\; \Sigma\,. \qquad \blacksquare$$

Lemma 3. If, for pairs $\Big\langle\, \gamma_1\,,\, \mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma_1)\, \Big\rangle$ and $\Big\langle\, \gamma_2\,,\, \mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma_2)\, \Big\rangle$ from $\mathrm{RP}_\sigma(\mathfrak{G})$, the representation-tuple $\mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma_2)$ is a reordering of representation-tuple $\mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma_1)$, then

$$\gamma_1 \;=\; \gamma_2\,. \qquad \blacksquare$$

Note that a particular $\gamma \in \Gamma_\sigma(\mathfrak{G})$ may also have several *substantially* different representations (in addition to those obtained by its reorderings), i.e. those involving different generators from $\mathfrak{G}$.

Remark 3 (a struct versus its generator representation). Thus, not only does the generator representation of a $\langle\sigma,\mathfrak{G}\rangle$-regular process *uniquely* define the corresponding struct γ, but it also provides *additional* formative (and useful[28]) information about the resulting struct, i.e. the process as evolved via a concrete sequence of generators versus the process as simply the finished struct. ◗

Lemma 4. If $\sigma_1 \sim \sigma_2$, then, for any family of abstract generators $\mathfrak{G}$,

$$\Big\{ [\![\gamma]\!] \;\Big|\; \gamma \in \Gamma_{\sigma_1}(\mathfrak{G}) \Big\} \;=\; \Big\{ [\![\gamma]\!] \;\Big|\; \gamma \in \Gamma_{\sigma_2}(\mathfrak{G}) \Big\}. \qquad \blacksquare$$

Before moving on to the next section, where we address this issue in greater detail, we note that the concept of generator representation $\mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma)$ embodies, in some sense, an incomplete form of structural representation, since the *mechanism* that produces the corresponding sequence of generators is not revealed: a more realistic (and useful) form of representation would clarify the underlying mechanism responsible for the generation of γ (as a member of some class $\mathfrak{C}$, including the selection—or, more accurately, *production*—of particular generators).

[28] From the point of view of the inductive learning (information) process (see also the next section).

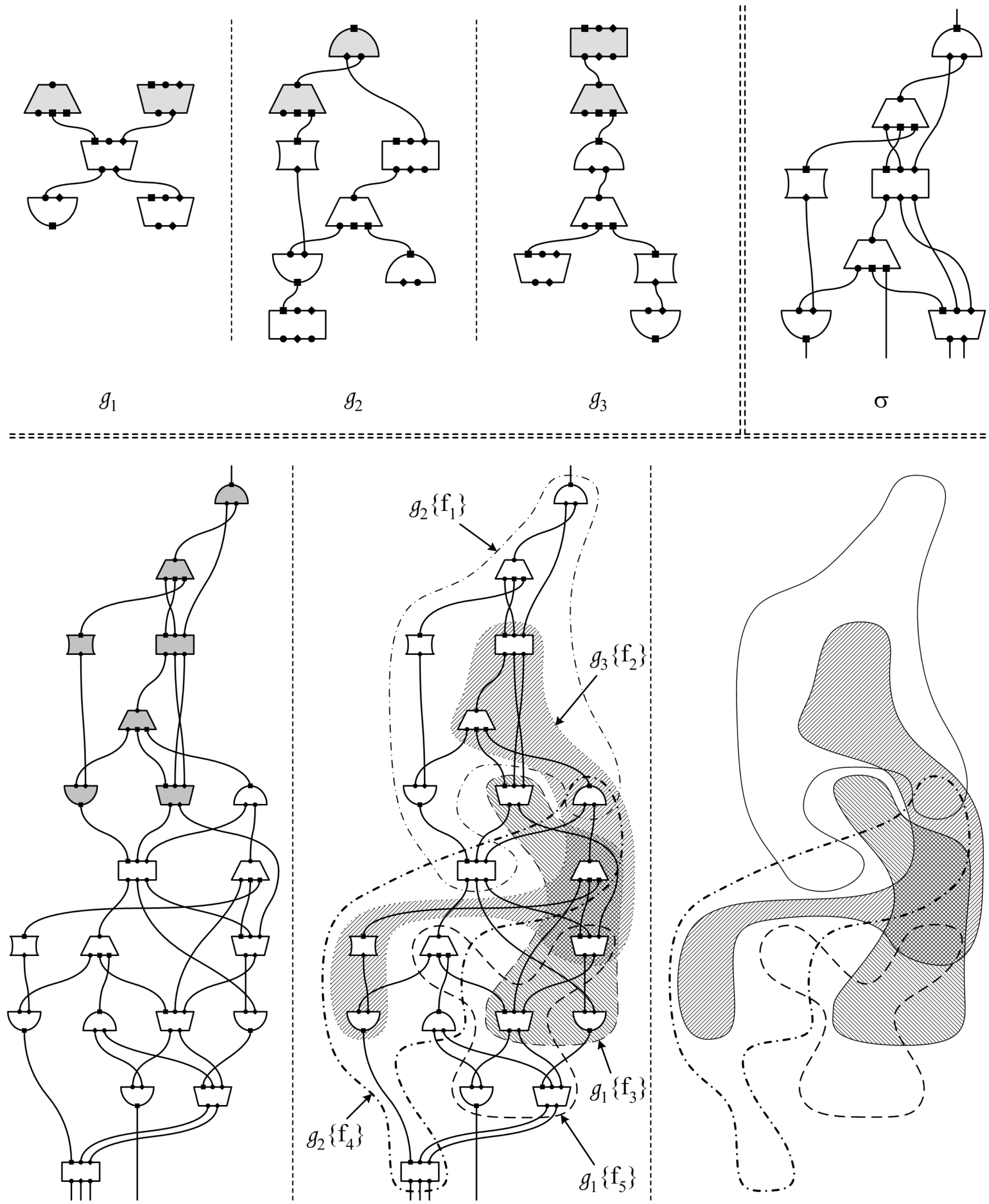

Figure 17: Top: A family $\mathfrak{G}$ consisting of three abstract generators (left), a "starting" struct σ (right). (Note that equality of context sizes is incidental.) Bottom: An example of a $\langle\sigma,\mathfrak{G}\rangle$-regular process $\sigma \lhd \mathfrak{g}_2\{f_1\} \lhd \mathfrak{g}_3\{f_2\} \lhd \mathfrak{g}_1\{f_3\} \lhd \mathfrak{g}_2\{f_4\} \lhd \mathfrak{g}_1\{f_5\}$ from set $\mathrm{RP}_\sigma(\mathfrak{G})$ illustrated in three different ways: simply as a struct with starting struct σ shaded (left), as a struct with generators identified (center), and as a combination of "generator outlines" (right). (The complete coverage of struct σ by generators is incidental.)

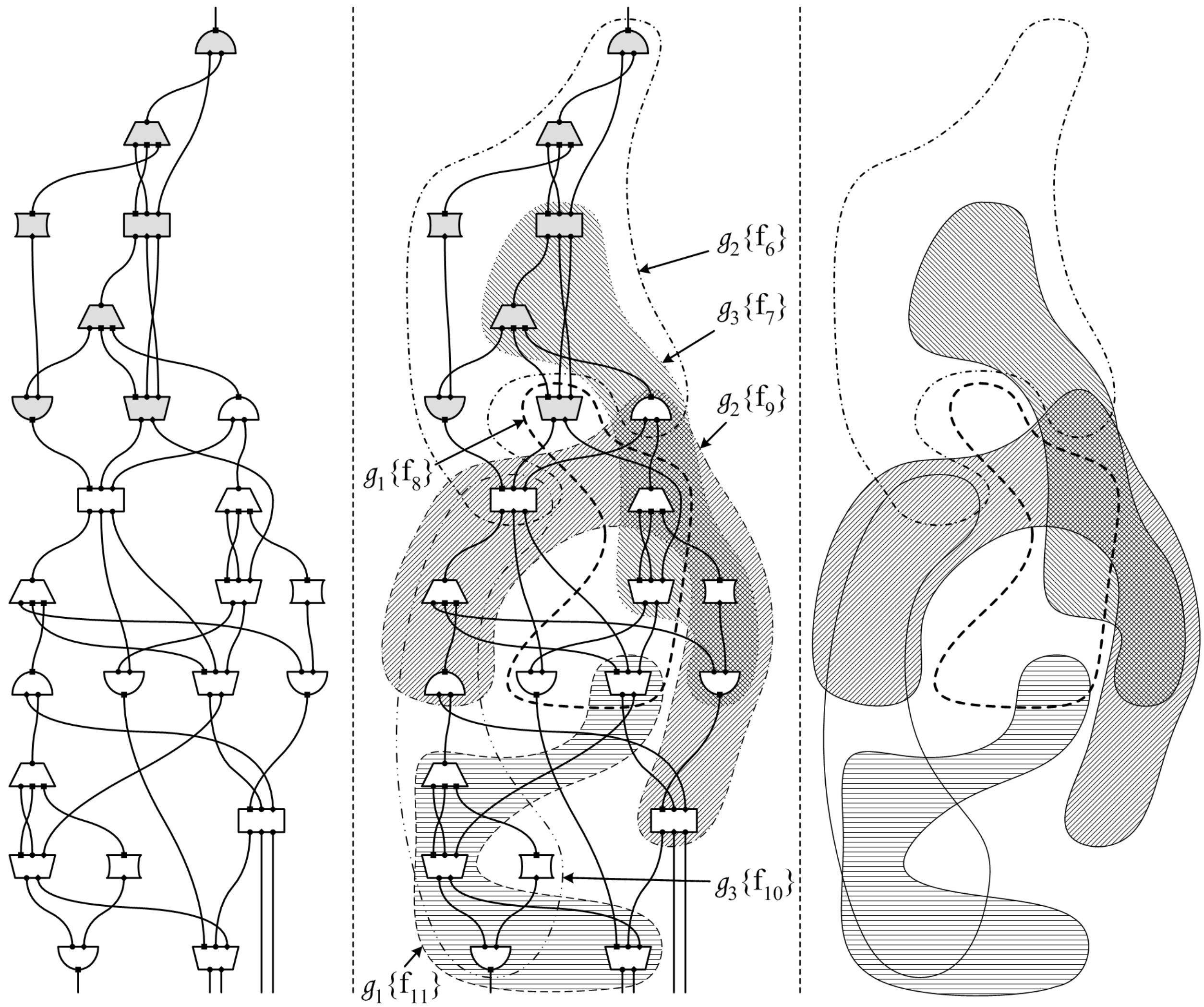

Figure 18: Another $\langle\sigma, \mathfrak{G}\rangle$-regular process $\sigma \triangleleft \mathfrak{g}_2\{f_6\} \triangleleft \mathfrak{g}_3\{f_7\} \triangleleft \mathfrak{g}_1\{f_8\} \triangleleft \mathfrak{g}_2\{f_9\} \triangleleft \mathfrak{g}_3\{f_{10}\} \triangleleft \mathfrak{g}_1\{f_{11}\}$ from the set $\mathrm{RP}_\sigma(\mathfrak{G})$ described in Fig. 17 (top).

6 Classes of structural processes

In this section, we briefly address the nature of *classes of structural processes*, which, of course, should also clarify the nature of primal classes (Def. 1). Unfortunately, this important section is less complete than is desirable, and in view of its density, we recommend that it be read at a slower pace. As was mentioned in the Introduction, the concept of class outlined here is much closer to that emerging from the recent research in evolutionary developmental biology[29], in which such processes as those implemented by regulatory (Hox) genes [65] play a central role, in contrast to processes more popular in classical biology and pattern recognition (i.e. in contrast to feature-based classes): “The embryo does not contain a description of the animal to which it will give rise, rather it contains a generative program for making it.

[29] Three popular references are [53], [51], and [52], and two of the standard ones are [64] and [65].

…There are thus no genes for 'arm' or 'leg' as such, but specific genes which become active during their formation." [53, pp. 199–200]

Moreover, the entire embryo development process (starting from a single fertilized egg cell) could serve as a suggestive example of a physical embodiment of the class generating process postulated below.

In our over-simplified picture of a regular structural process, for a struct σ and a finite family $\mathfrak{G}$ of abstract generators with a non-empty set of $\langle\sigma, \mathfrak{G}\rangle$-regular processes $\mathrm{RP}_\sigma(\mathfrak{G})$, we want to consider a class of structural processes closely related to the set $\mathrm{RP}_\sigma(\mathfrak{G})$ that have *strong structural similarity* to each other. The following postulate addresses the concept of a *class* $\mathfrak{C}$ of structural processes: the processes in each such class are generated by a 'class generating system'. We do expect this postulate to be gradually refined in the near future until it reaches the status of a definition.

However, before moving on to our Class Representation Postulate, it is important to outline, at least very briefly, the overall *development* of a structural process as a member of a class of structural processes. By analogy with organisms, one can view the overall development of such processes as consisting of two (not necessarily sharply separated) stages: the maturation and mature stages. The maturation stage is analogous to the embryonic stage and is usually "hidden" inside the body of the corresponding transformation (introduced in the next section). The mature stage corresponds to the adult stage of an organism, i.e. when the process achieves (its) "regularity" and thus becomes *more* accessible for interaction (with other processes)[30]. In our preliminary definition of a regular structural process (Def. 14), what we've *formally defined as a regular process refers only to an initial, overly simple description of the mature stage of a structural process.*

Class Representation Postulate. A **class** $\mathfrak{C}$ of (structural) processes whose set of **(class) contexts** is $\Sigma_{\mathfrak{C}}$, $\Sigma_{\mathfrak{C}} \subseteq \Sigma$, is defined as

$$\mathfrak{C} \;=\; \bigcup_{\sigma \,\in\, \Sigma_{\mathfrak{C}}} \mathfrak{C}^\sigma \,,$$

where each $\mathfrak{C}^\sigma$—called a **class of (structural) processes for context** σ—is defined, or specified, by $\mathfrak{R}\langle\mathfrak{C}^\sigma\rangle$, called a **$\sigma$-dependent class representation**.

Such a representation $\mathfrak{R}\langle\mathfrak{C}^\sigma\rangle$ could be thought of as an abstract "parametric" *specification* of $\mathfrak{C}^\sigma$; it works together with the primordial and universal blueprint for class generation—denoted $\mathscr{GS}$ and called a **universal class generating system**—to produce the class elements. More specifically, when a representation $\mathfrak{R}\langle\mathfrak{C}^\sigma\rangle$ is supplied to $\mathscr{GS}$, the resulting, or instantiated, system $\mathscr{GS}\big(\mathfrak{R}\langle\mathfrak{C}^\sigma\rangle\big)$—called the **generating system for class** $\mathfrak{C}^\sigma$—*at some point*[31] starts producing those, and only those, regular processes that are associated with class $\mathfrak{C}$. To simplify notation, we denote $\mathscr{GS}\big(\mathfrak{R}\langle\mathfrak{C}^\sigma\rangle\big)$ by $\mathscr{GS}(\mathfrak{C}^\sigma)$ and then define $\mathfrak{C}^\sigma$ as a pair

$$\mathfrak{C}^\sigma \;=\; \Big\langle\, \mathscr{GS}(\mathfrak{C}^\sigma)\,,\, \mathrm{SR}(\mathfrak{C}^\sigma)\,\Big\rangle\,,$$

[30] It is important to emphasize that the maturation stage is also accessible for interaction, but to a lesser extent, though such "interactions" are prone to have a *greater effect on the emerging structure* of the process.

[31] See the discussion (immediately preceding the postulate) about the two stages in the development of a structural process as a member of a class, where the onset of the second stage is marked, in our simplified approach, by the solidification of the *final* family $\mathfrak{G}$ of abstract generators.

where $\mathrm{SR}(\mathfrak{C}^\sigma)$ is a set of "structurally regular" processes outputted by $\mathscr{GS}(\mathfrak{C}^\sigma)$.[32]

Moreover, $\mathscr{GS}(\mathfrak{C}^\sigma)$ is a *hierarchical* generating system involving a few levels: as in a developmental process, a *single* entity at a given level gradually gives rise to a structural organization of *several* entities at the lower level (in particular, an application of a second-lowest-level generator yields a structural organization of lowest-level primitives). [The *structure* of such class generating systems holds the main key to understanding the concept of 'similarity' between a structural process and a class.]

The previous concept of a regular process $\left\langle \gamma\,,\, \mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma) \right\rangle$ (see Def. 14) should now be substantially revised to replace its second component by $\mathfrak{R}\langle\mathfrak{C}^\sigma\rangle(\gamma)$. Hence, for each struct γ, $\gamma \in \mathrm{SR}(\mathfrak{C}^\sigma)$, outputted by $\mathscr{GS}(\mathfrak{C}^\sigma)$, one can also introduce the concept of the **class representation of** γ, $\mathscr{GS}(\gamma\,,\mathfrak{C}^\sigma)$, which is a generalization of $\mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma)$.

Finally, for two different classes $\mathfrak{C}_1$ and $\mathfrak{C}_2$, and two of their structs $\gamma_1 \in \mathrm{SR}(\mathfrak{C}_1^{\sigma_1})$ and $\gamma_2 \in \mathrm{SR}(\mathfrak{C}_2^{\sigma_2})$, we postulate that

$$\mathscr{GS}(\gamma_1\,,\mathfrak{C}_1^{\sigma_1}) \;\; \neq \;\; \mathscr{GS}(\gamma_2\,,\mathfrak{C}_2^{\sigma_2})\,.$$

At the same time, it is also important to note that, given a class and its subclass, their generating systems have to be consistent in the following way: the subclass generating system must be a refinement, or elaboration, of the class generating system. ◗

Note that, as a substantial refinement of the generator representation $\mathcal{R}\langle\sigma,\mathfrak{G}\rangle(\gamma)$ (see the Remark on p. 35), the representation $\mathscr{GS}(\gamma\,,\mathfrak{C}_\sigma)$ contains *further*, hierarchically organized, representational information regarding struct γ. In other words, the unfolding of a structural process is not controlled by a fixed set $\mathfrak{G}$ of generators, but rather by the logic of the class generating system, which, in turn, dynamically produces/modifies the "current" set $\mathfrak{G}$.

Thus, what distinguishes a segment of a more "realistic" structural process from a $\langle\sigma,\mathfrak{G}\rangle$-regular process is the presence in that segment of additional *generative* structure from the corresponding class representation. The role of this additional structure becomes clearer when we consider the role it plays in the inductive learning process: a particular class representation, $\mathfrak{R}\langle\mathfrak{C}_\sigma\rangle$, must be inductively learned based on the training structs and the (fixed) universal class generating system $\mathscr{GS}$ by directly or indirectly endowing the training structs with an optimal hierarchical structure. In other words, the learning process attempts to recover, relying on $\mathscr{GS}$, the higher levels of the hierarchy mentioned in the above postulate based on the lower levels, starting with observed structs (see also Section 9).

Speculative remark. One of the anticipated consequences of the above postulate is that it should facilitate the introduction of an analogue of the concept of a *topology* for a class (i.e. how 'close') and also facilitates the emergence of a "class topography" (i.e. how 'typical'). Moreover, it is natural to assume that a fixed set of generators does not uniquely induce

[32] Obviously, it would have been sufficient to define $\mathfrak{C}^\sigma$ to be simply $\mathscr{GS}(\mathfrak{C}^\sigma)$, since the generating system "knows" what it produces; however, given the not sufficiently formal nature of the specification of $\mathscr{GS}(\mathfrak{C}^\sigma)$, the above definition as a pair might be helpful for expository purposes. Note that it is $\mathscr{GS}\Big(\mathfrak{R}\langle\mathfrak{C}^\sigma\rangle\Big)$ that actually "runs the show (for the class)". Also note that $\mathrm{SR}(\mathfrak{C}^\sigma)$ is, in fact, a generalization of $\mathrm{RP}_\sigma(\mathfrak{G})$.

the topology of a class: it is the overall structure, or "structural features", of the generative system that should ensure uniqueness. The main difficulty in refining the above class representation postulate to include the latter considerations is due to the fact that no one has ever had any experience with introducing such a generative mechanism in the structural setting outlined by this formalism. ◗

For a very early attempt to implement the concept of generativity via Markov stochastic processes in the ETS formalism, see [2], [39], [40] (where this was accomplished in a more conventional manner by attaching numeric weights to transforms, which were treated as earlier analogues of generators). Obviously, we could have introduced the above class definition in a similar manner by associating a weight with each generator and invoking the above Markov process model, but our reluctance to follow such a numerically inspired approach here is due to the desire not to trivialize the above scheme. Again, this *by no means* is to say that early applications must discount such an approach at this initial stage; in fact, being faced with some immediate application, one should definitely *consider* this option.

In what follows, since the above postulate is not yet formally realized, of necessity, we blur the distinction between the above regular process (Section 5) and the more "realistic" structural process from a class $\mathfrak{C}$.

7 Transformations

In this section, we introduce two central concepts (Def. 16), i.e. abstract and concrete transformations, which should clarify those of abstract and concrete primitives. As was mentioned in the Introduction, by a 'transformation' we mean a macroevent that is responsible for transforming one set of "adjacent" structural processes into another set of processes. Thus, a transformation entrenches the nature of a particular (and possibly recurring) *pattern of interactions for the 'initial' structural processes which results in modified 'terminal' structural processes*: it accomplishes this by producing terminal class representations from the representations of the initial classes. An abstract transform, then, can be viewed as an apparatus responsible for generating *new/modified classes of processes* out of existing ones. Moreover, as was suggested just before the Class Representation Postulate, the maturation stage of a resulting terminal structural process is "hidden" inside the body of the corresponding transform.

As was mentioned above, we blur the distinction between the above (our) regular processes and the more realistic structural process from a class $\mathfrak{C}$, as discussed in the last section. This over-simplification allows us to introduce the concept of a transformation in a more formal manner, and at the same time facilitates an easier initial grasp of this concept.

Assumption 1. In what follows we assume that, for the basic representational level, a small set $\mathscr{C}$ of classes of regular processes has been specified:

$$\mathscr{C} = \left\{ \mathfrak{C}_1 , \mathfrak{C}_2 , \dots , \mathfrak{C}_{m'} \right\} . \qquad ◗$$

Before introducing transformations, we need the following auxiliary definition.

<u>Definition</u> 15. A p-tuple of regular process classes selected from $\mathscr{C}$

$$\langle\, \mathfrak{C}_{i_1}\,,\, \mathfrak{C}_{i_2}\,,\, \ldots\,,\, \mathfrak{C}_{i_p}\,\rangle$$

is called a **tuple of classes** (TC). Moreover, a p-tuple

$$\langle\, \mathfrak{c}_{i_1}\,,\, \mathfrak{c}_{i_2}\,,\, \ldots\,,\, \mathfrak{c}_{i_p}\,\rangle\,,$$

such that

$$\mathfrak{c}_{i_j} \in \mathfrak{C}_{i_j} \quad \text{and} \quad \mathfrak{c}_{i_j} \neq \mathfrak{c}_{i_k} \;\; \text{for} \;\; j \neq k\,,$$

is called a **tuple of class elements** (TCE) associated with TC.

We will need to distinguish between *two kinds of tuples* of regular process classes, which we will denote as ITC (**initial tuple**) and TTC (**terminal tuple**), and the corresponding tuples of class elements will be denoted ITCE and TTCE, respectively.

Given a TC, the corresponding **set of all tuples of class elements** is denoted $\mathbf{TCE}_{\,\mathrm{TC}}$, $\mathbf{TCE}_{\,\mathrm{TC}} \subseteq \mathfrak{C}_{i_1} \times \ldots \times \mathfrak{C}_{i_p}$. ▶

Moving on to our central definition and relying on the terminology of Def. 15, we note that the following concept of a transformation in fact *designates* a particular kind of interaction[33] *between* the constituent processes of ITC, resulting in the generation of the terminal tuple of regular processes TTC.

<u>Definition</u> 16. For a given **abstract transformation name** $\widehat{\boldsymbol{\tau}}$, the corresponding **abstract transformation** of regular processes (associated with $\mathscr{C}$), or simply **abstract transform**, is an enumerable set $\boldsymbol{\tau}$[34]

$$\boldsymbol{\tau} \;=\; \Big\{\, \boldsymbol{\tau}(\mathrm{ITCE}) \;\Big|\; \mathrm{ITCE} \in \mathbf{TCE}_{\,\mathrm{ITC}(\widehat{\boldsymbol{\tau}})} \Big\}$$

whose generic element $\boldsymbol{\tau}(\mathrm{ITCE})$, called a *corresponding* (concrete) **transform**, is defined as follows:[35]

for each ITCE from $\mathbf{TCE}_{\,\mathrm{ITC}(\widehat{\boldsymbol{\tau}})}$ the corresponding transform is defined as[36]

$$\boldsymbol{\tau}(\mathrm{ITCE}) \;=\; \tau_{\mathrm{ITCE}} \;\stackrel{\text{def}}{=}\; \Big\langle\, \widehat{\boldsymbol{\tau}}\,,\, \mathrm{ITC}(\widehat{\boldsymbol{\tau}})\,,\, \mathrm{TTC}(\widehat{\boldsymbol{\tau}})\,,\, \mathrm{ITCE}\,,\, B_{\boldsymbol{\tau}(\mathrm{ITCE})} \Big\rangle\,,$$

where

[33] At present, given our limited applied experience, we do not propose any classification of these interactions (into concrete categories).

[34] Note that, here, the set $\mathbf{TCE}_{\,\mathrm{ITC}(\boldsymbol{\tau})}$ plays a role analogous to that of the set of labels $\mathcal{L}$ associated with a primitive $\boldsymbol{\pi}$ (see Def. 1). The latter important analogy will be exploited in the next section, when transforms are replaced by next-level primitives (see mapping $apt_{\,i\text{-lab}}$ in Def. 17).

[35] As in Def. 1, both of the following notations, i.e. $\boldsymbol{\tau}(\mathrm{ITCE})$ and τ_{ITCE}, will be used.

[36] For simplicity, the hat accent on $\widehat{\boldsymbol{\tau}}$ is dropped when it is used as a lower index.

$$\mathrm{ITC}(\widehat{\boldsymbol{\tau}}) = \langle\, \mathfrak{C}^{\mathrm{Init}}_{\boldsymbol{\tau},1}, \ldots, \mathfrak{C}^{\mathrm{Init}}_{\boldsymbol{\tau},p} \,\rangle \qquad \mathfrak{C}^{\mathrm{Init}}_{\boldsymbol{\tau},i} \text{ is the (given) } i^{\,\mathbf{th}} \textbf{ initial class for } \widehat{\boldsymbol{\tau}},\ p > 0$$

$$\mathrm{ITCE} = \langle\, \overline{\mathfrak{c}_1}, \ldots, \overline{\mathfrak{c}_p} \,\rangle \qquad \overline{\mathfrak{c}_i} \text{ is the } i^{\,\mathbf{th}} \textbf{ initial class element} \text{ for } \boldsymbol{\tau}(\mathrm{ITCE}),\ \overline{\mathfrak{c}_i} \in \mathfrak{C}^{\mathrm{Init}}_{\boldsymbol{\tau},i}$$

$$\mathrm{TTC}(\widehat{\boldsymbol{\tau}}) = \langle\, \mathfrak{C}^{\mathrm{Term}}_{\boldsymbol{\tau},1}, \ldots, \mathfrak{C}^{\mathrm{Term}}_{\boldsymbol{\tau},q} \,\rangle \qquad \mathfrak{C}^{\mathrm{Term}}_{\boldsymbol{\tau},j} \text{ is the (given) } j^{\,\mathbf{th}} \textbf{ terminal class for } \widehat{\boldsymbol{\tau}}$$

and

$B_{\boldsymbol{\tau}(\mathrm{ITCE})}$ is a set of structs satisfying

$$\forall\, \beta \in B_{\boldsymbol{\tau}(\mathrm{ITCE})} \quad \exists\, \langle\, \underline{\mathfrak{c}_1}, \ldots, \underline{\mathfrak{c}_q} \,\rangle \in \mathbf{TCE}_{\,\mathrm{TTC}(\widehat{\boldsymbol{\tau}})} \quad \text{such that, introducing}$$

$$\overline{\mathfrak{c}_i} = \big\langle\, \overline{\gamma_i}, \mathfrak{R}\langle\, \overline{\sigma_i}, \overline{\mathfrak{C}^{\mathrm{Init}}_{\boldsymbol{\tau},i}} \,\rangle \big\rangle \text{ and } \underline{\mathfrak{c}_j} = \big\langle\, \underline{\gamma_j}, \mathfrak{R}\langle\, \underline{\sigma_j}, \underline{\mathfrak{C}^{\mathrm{Term}}_{\boldsymbol{\tau},j}} \,\rangle \big\rangle,$$

we have:

(i) structs $\overline{\gamma_1}, \ldots, \overline{\gamma_p}, \beta, \underline{\gamma_1}, \ldots, \underline{\gamma_q}$ are unifiable

(ii) $\mathrm{Init}(\, \overline{\gamma_1} \Cup \ldots \Cup \overline{\gamma_p} \Cup \beta \,) \cap \mathrm{Init}(\beta) = \varnothing$

(iii) $\mathrm{Term}(\, \beta \Cup \underline{\gamma_1} \Cup \ldots \Cup \underline{\gamma_q} \,) \cap \mathrm{Term}(\beta) = \varnothing\,.$

We denote by $\mathbf{T}_{\boldsymbol{\Pi},\mathscr{C}}$, or simply $\mathbf{T}$, the set of all abstract transforms associated with $\mathscr{C}$, and by $\mathrm{T}_{\boldsymbol{\Pi},\mathscr{C}}$, or simply T, the set of all corresponding (concrete) transforms. ▶

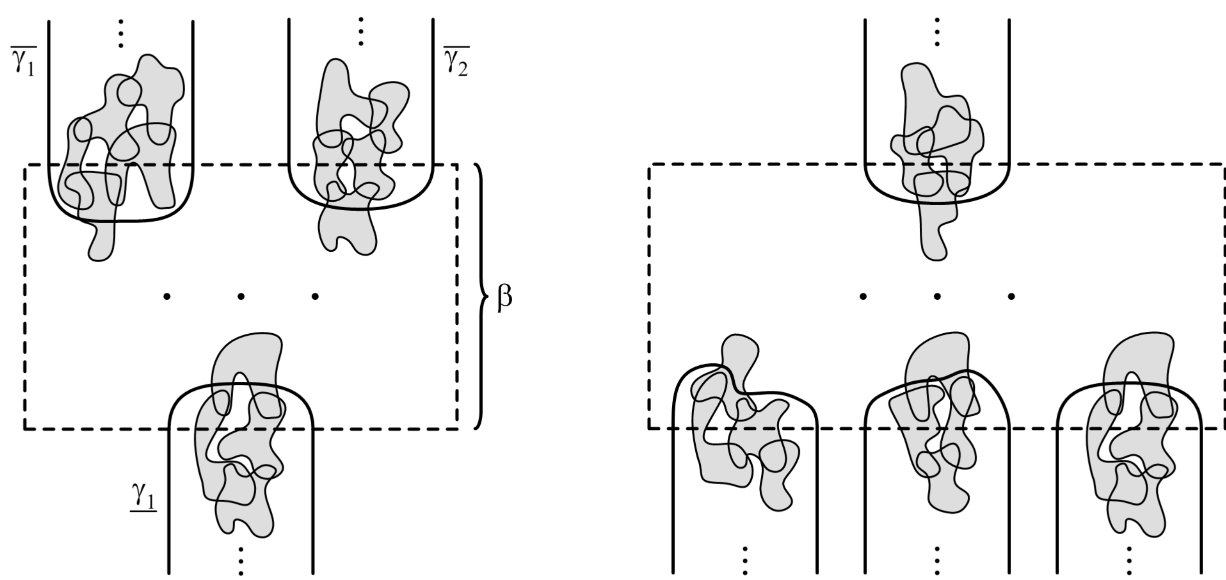

Figure 19: An illustration of two concrete transforms in which the bodies and regular processes are only partially shown.

It is important to stress that, as will be seen from the next section, the present form of the ETS formalism does not substantially depend on the great accuracy of information encapsulated in the body of the transform. Thus, in particular, it is easy to see that there exist many perturbations of body β of transform τ (including those associated with a substantial noise component) that can still serve as legal bodies for the "same" transform. However, as was stressed several times above, in the future the body of a transform will be a more complex object than a struct, since it has to account for the appropriate class transforming apparatus.

Remark 4 (on the relationship between transforms and their processes). When observing two transforms directly connected by some regular process, it is useful to note that, if a particular initial process of the top transform is *structurally identical* (see Def. 8) to the connecting process, this means that the body of the top transform is actually responsible for *"restarting"* this regular process. Also, when dealing, for example, with a primitive transform whose initial and terminal tuples are identical, one might easily misunderstand the situation as "processes passing unaffected through a primitive", leading to the wrong idea of processes being "static objects", rather than temporal entities. In general (unless evidence suggests otherwise), if an initial regular process is observed "passing through", i.e. non-interacting with other processes, then when constructing the transform this process *should not be treated as part of the transform.* ◗

We are now ready to introduce one of the most radical (and unique) features of the proposed formalism, the *natural* emergence of the next level of representation.

8 Multi-level inductive structures

In this section, we shall see why the ETS formalism allows a natural transition to the next level of representation. Such a transition consists of the construction of a new (next-level) set of primitives, which can then be used to introduce next-level structs, transforms, etc. in the above manner.

Level ascension principle. The pattern of class interactions corresponding to a transform may be adequately captured at the next representational level by a new primitive whose initials and terminals are obtained by suppressing the structure of the transform's initial and terminal classes, and by suppressing the internal structure of its bodies (in the manner described below, in Def. 17). ►

Notational convention 3. From this section onwards, next-level notations will be denoted with the addition of a prime ($'$) to the corresponding present-level notation. Moreover, the *lower (subscript) index of a mapping name* refers to the codomain of the mapping. ◗

We now move to the stage in which the agent has already identified (i.e. constructed) a set of transformations relevant to the present observations.

Assumption 2. Having identified a set $\mathscr{C}$ of classes of regular processes in Assumption 1, we further assume that a set $\mathbf{TS}_{\mathscr{C}}$, or simply $\mathbf{TS}$, of abstract transforms,

$$\mathbf{TS}_{\mathscr{C}} \;=\; \{\, \boldsymbol{\tau}_1,\, \boldsymbol{\tau}_2,\, \ldots,\, \boldsymbol{\tau}_{n'} \,\},$$

called a **transformation system**, is also identified. Recall that a generic element $\boldsymbol{\tau}_i(\mathrm{ITCE})$ of $\boldsymbol{\tau}_i$ is

$$\boldsymbol{\tau}_i(\mathrm{ITCE}) \;=\; \Big\langle\, \widehat{\boldsymbol{\tau}}_i\,,\, \mathrm{ITC}(\widehat{\boldsymbol{\tau}}_i)\,,\, \mathrm{TTC}(\widehat{\boldsymbol{\tau}}_i)\,,\, \mathrm{ITCE}\,,\, B_{\boldsymbol{\tau}_i(\mathrm{ITCE})} \,\Big\rangle,$$

where

$$\begin{aligned} \mathrm{ITC}(\widehat{\boldsymbol{\tau}}_i) &= \langle\, \mathfrak{C}^{\mathrm{Init}}_{\boldsymbol{\tau}_i,1}\,,\, \ldots\,,\, \mathfrak{C}^{\mathrm{Init}}_{\boldsymbol{\tau}_i,p(i)} \,\rangle \\ \mathrm{TTC}(\widehat{\boldsymbol{\tau}}_i) &= \langle\, \mathfrak{C}^{\mathrm{Term}}_{\boldsymbol{\tau}_i,1}\,,\, \ldots\,,\, \mathfrak{C}^{\mathrm{Term}}_{\boldsymbol{\tau}_i,q(i)} \,\rangle\,. \end{aligned}$$

◗

The following definition introduces the concept of next-level primitive transformation (both abstract and concrete). Since the structure of the next definition follows that of Def. 1, it might be useful to review it at this point.

Definition 17. For the above transformation system $\mathbf{TS}_{\mathscr{C}}$, the set $\mathbf{C}'$ of **next-level classes** is defined as follows:

$$\mathbf{C}' \;\stackrel{\mathrm{def}}{=}\; \{\, C'_1,\, C'_2,\, \ldots,\, C'_{m'} \,\},$$

where $|C'_i| \;=\; |\mathfrak{C}_i|\,,\ 1 \le i \le m'$, i.e. we have a bijection

$$ap_{\,i\text{-class}} \colon \mathfrak{C}_i \to C'_i$$

but the *structure* of each element in $\mathfrak{C}_i$ is abstracted away, hence the name "abstraction of processes" (ap). The set of **names of next-level primitives** is:

$$\widehat{\boldsymbol{\Pi}'} \;\stackrel{\mathrm{def}}{=}\; \{\, \widehat{\boldsymbol{\tau}_1},\, \widehat{\boldsymbol{\tau}_2},\, \ldots,\, \widehat{\boldsymbol{\tau}_{m'}} \,\},$$

i.e. $\widehat{\boldsymbol{\pi}'_i} \;=\; \widehat{\boldsymbol{\tau}}_i$. As in Def. 1, before defining $\boldsymbol{\pi}'_i$, we need to introduce the following three concepts. For each $\widehat{\boldsymbol{\pi}'_i}$, define the **tuple of initial next-level classes**

$$\mathrm{Init}(\widehat{\boldsymbol{\pi}'_i}) \;\stackrel{\mathrm{def}}{=}\; \langle\, C'_{j_1}\,,\, C'_{j_2}\,,\, \ldots\,,\, C'_{j_{p(i)}} \,\rangle,$$

where

$$C'_{j_k} \;=\; ap_{\,i\text{-class}}\big(\, \mathfrak{C}^{\mathrm{Init}}_{\boldsymbol{\tau}_i,\,j_k} \,\big) \qquad\qquad 1 \le k \le p(i)\,.$$

Then, the **set of next-level labels associated with** $\boldsymbol{\pi}'_i$ is

$$\mathcal{L}'_i \;\stackrel{\mathrm{def}}{=}\; C'_{j_1} \times\, C'_{j_2} \times \ldots \times C'_{j_{p(i)}}$$

and we have a bijection

$$apt_{\,i\text{-lab}} \colon \mathbf{TCE}_{\,\mathrm{ITC}(\widehat{\boldsymbol{\tau}}_i)} \to \mathcal{L}'_i$$

$$apt_{\,i\text{-lab}} \;\stackrel{\mathrm{def}}{=}\; ap_{\,j_1\text{-class}} \times ap_{\,j_2\text{-class}} \times \ldots \times ap_{\,j_{p(i)}\text{-class}}\,,$$

hence the name "abstraction of process tuples" (see Defs. 15 and 16). The concept of $\mathrm{Term}(\widehat{\boldsymbol{\pi}'_i})$ is defined in a manner similar to $\mathrm{Init}(\widehat{\boldsymbol{\pi}'_i})$.

Define $\boldsymbol{\pi}'_i$ as a set

$$\boldsymbol{\pi}'_i \;=\; \{\, \boldsymbol{\pi}'_i(a') \mid a' \in \mathcal{L}'_i \,\}$$

whose generic element $\boldsymbol{\pi}'_i(a')$ is:[37]

$$\forall\, a' \in \mathcal{L}'_i \qquad \boldsymbol{\pi}'_i(a') \;=\; \pi'_{ia'} \;\stackrel{\mathrm{def}}{=}\; \langle\, \widehat{\boldsymbol{\pi}'_i},\ \mathrm{Init}(\widehat{\boldsymbol{\pi}'_i}),\ \mathrm{Term}(\widehat{\boldsymbol{\pi}'_i}),\ a' \,\rangle\,.$$

see Fig. 20. Set $\boldsymbol{\pi}'_i$ is called a **next-level abstract primitive transformation**, or simply **next-level abstract primitive**, and its generic element $\pi'_{ia'}$ is called a **corresponding** (concrete) **next-level primitive**. We denote by $\boldsymbol{\Pi}'$ the finite set of all next-level abstract primitives $\boldsymbol{\pi}'_i$, $1 \le i \le n$ and by Π' the set of all (concrete) next-level primitives. ▶

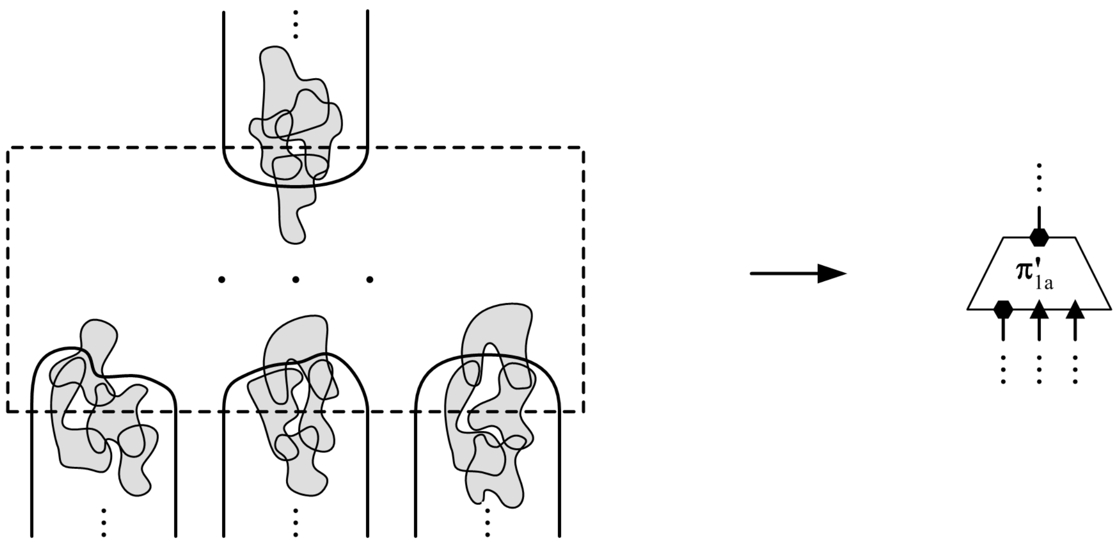

Figure 20: Transition from a transform (left) to the corresponding next-level primitive (right).

Since the structure of a next-level primitive is absolutely identical to that of the initial-level primitive, all remaining initial-level concepts are immediately applicable.

Definition 18. Next-level analogues of definitions 2 through 16 follow immediately. ▶

We are now ready to state a formal version of the above level ascension principle, which might be viewed as a convenient encapsulation of the main concepts in Def. 17.

Correspondence mappings between consecutive levels. For the above transformation system $\mathbf{TS}_{\mathscr{C}}$,

$$\mathbf{TS}_{\mathscr{C}} \;=\; \{\, \boldsymbol{\tau}_1,\, \boldsymbol{\tau}_2,\, \ldots,\, \boldsymbol{\tau}_{n'} \,\}\,,$$

[37] As in Def. 1, both of the following notations, i.e. $\boldsymbol{\pi}'_i(a')$ and $\pi'_{ia'}$, will be used.

the corresponding **consecutive level mappings** are the following three bijective mappings:

(i) $$\mathbb{CM}_{\,i\text{-class}}\colon \mathfrak{C}_i \to C'_i \qquad 1 \le i \le m'$$
$$\mathbb{CM}_{\,i\text{-class}} \stackrel{\text{def}}{=} ap_{\,i\text{-class}}\,,$$

(ii) $$\mathbb{CM}_{\text{a-prim}}\colon \mathbf{TS}_{\mathscr{C}} \to \boldsymbol{\Pi}'$$
$$\mathbb{CM}_{\text{a-prim}}\big(\,\boldsymbol{\tau}_i\,\big) \stackrel{\text{def}}{=} \boldsymbol{\pi}'_i\,,$$

(iii) $$\mathbb{CM}_{\,i\text{-lab}}\colon \mathbf{TCE}_{\,\text{ITC}(\widehat{\boldsymbol{\tau}}_i)} \to \mathcal{L}'_i$$
$$\mathbb{CM}_{\,i\text{-lab}} \stackrel{\text{def}}{=} apt_{\,i\text{-lab}}\,.$$

►

It is not difficult to see how the above consecutive level correspondence can be extended to include the following mapping from concrete transforms to concrete primitives:

$$\mathbb{CM}_{\text{c-prim}}\colon \mathrm{T} \to \Pi'$$
$$\mathbb{CM}_{\text{c-prim}}\big(\,\boldsymbol{\tau}_i(\text{ITCE})\,\big) \;=\; \boldsymbol{\pi}'_i(\mathfrak{a}')\,,$$

where
$$\mathbb{CM}_{\text{a-prim}}\big(\,\boldsymbol{\tau}_i\,\big) \;=\; \boldsymbol{\pi}'_i \qquad\qquad \mathbb{CM}_{\,i\text{-lab}}\big(\,\text{ITCE}\,\big) \;=\; \mathfrak{a}'\,.$$

Finally, we can encapsulate the entire developed mathematical structure as a single concept in the following definition. Note that we substitute the arrow $\nearrow$ for the various $\mathbb{CM}$-mappings.

<u>Definition</u> 19. A **(single-level) inductive structure** is a pair

$$\langle\,\boldsymbol{\Pi}\,,\,\mathbf{TS}_{\mathscr{C}}\,\rangle\,,$$

where $\boldsymbol{\Pi}$ is a set of abstract primitives and $\mathbf{TS}_{\mathscr{C}}$ is a transformation system. However this pair will also encompass *all relevant concepts*, such as structs, regular processes, transforms, etc.

A **multi-level**, or more precisely an l-level, **inductive structure** $\mathcal{MIS}$ is an l-tuple[38]

$$\mathcal{MIS} \;=\; \big\langle\,\langle\,\boldsymbol{\Pi}\,,\,\mathbf{TS}\,\rangle\,,\,\langle\,\boldsymbol{\Pi}'\,,\,\mathbf{TS}'\,\rangle\,,\,\ldots\,,\,\langle\,\boldsymbol{\Pi}^{(l-1)}\,,\,\mathbf{TS}^{(l-1)}\,\rangle\,\big\rangle\,,$$

where $\mathbf{TS}^{(l-1)} \;=\; \varnothing$, $\langle\,\boldsymbol{\Pi}^{(k)}\,,\,\mathbf{TS}^{(k)}\,\rangle$ is the k^{th} level inductive structure, and the transition between two consecutive inductive structures is accomplished in the manner outlined above

[38] For simplicity, we drop the index $\mathscr{C}^{(k)}$ from $\mathbf{TS}^{(k)}_{\mathscr{C}^{(k)}}$.

in this section (see Figs. 21 and 22). For the k^{th} level inductive structure in $\mathcal{MIS}$ we will use the notation

$$
\begin{aligned}
\mathcal{MIS}(k) \;=\; \left\langle \mathbf{\Pi}^{(k)}, \mathbf{TS}^{(k)} \right\rangle & \qquad\qquad k = 0, 1 \ldots, l-1 \\
\mathfrak{C}_i^{(k)} \nearrow C_i^{(k+1)} & \qquad\qquad k = 0, 1 \ldots, l-2 \\
\boldsymbol{\tau}^{(k)} \nearrow \boldsymbol{\pi}^{(k+1)} & \qquad\qquad k = 0, 1 \ldots, l-2 \\
\mathrm{ITCE}^{(k)} \nearrow \boldsymbol{a}^{(k+1)} & \qquad\qquad k = 0, 1 \ldots, l-2 \\
\boldsymbol{\tau}^{(k)}(\mathrm{ITCE}^{(k)}) \nearrow \boldsymbol{\pi}^{(k+1)}(\boldsymbol{a}^{(k+1)}) \qquad \text{or} \qquad \tau^{(k)}_{\mathrm{ITCE}^{(k)}} \nearrow \pi^{(k+1)}_{\boldsymbol{a}^{(k+1)}} & \qquad\qquad k = 0, 1 \ldots, l-2\,,
\end{aligned}
$$

where the arrow $\nearrow$ stands for the appropriate $\mathbb{CM}$-mapping. ▶

In general, we expect that as levels emerge, the lower levels gradually rigidify, i.e. stabilize, and do so at a faster pace than the upper levels.

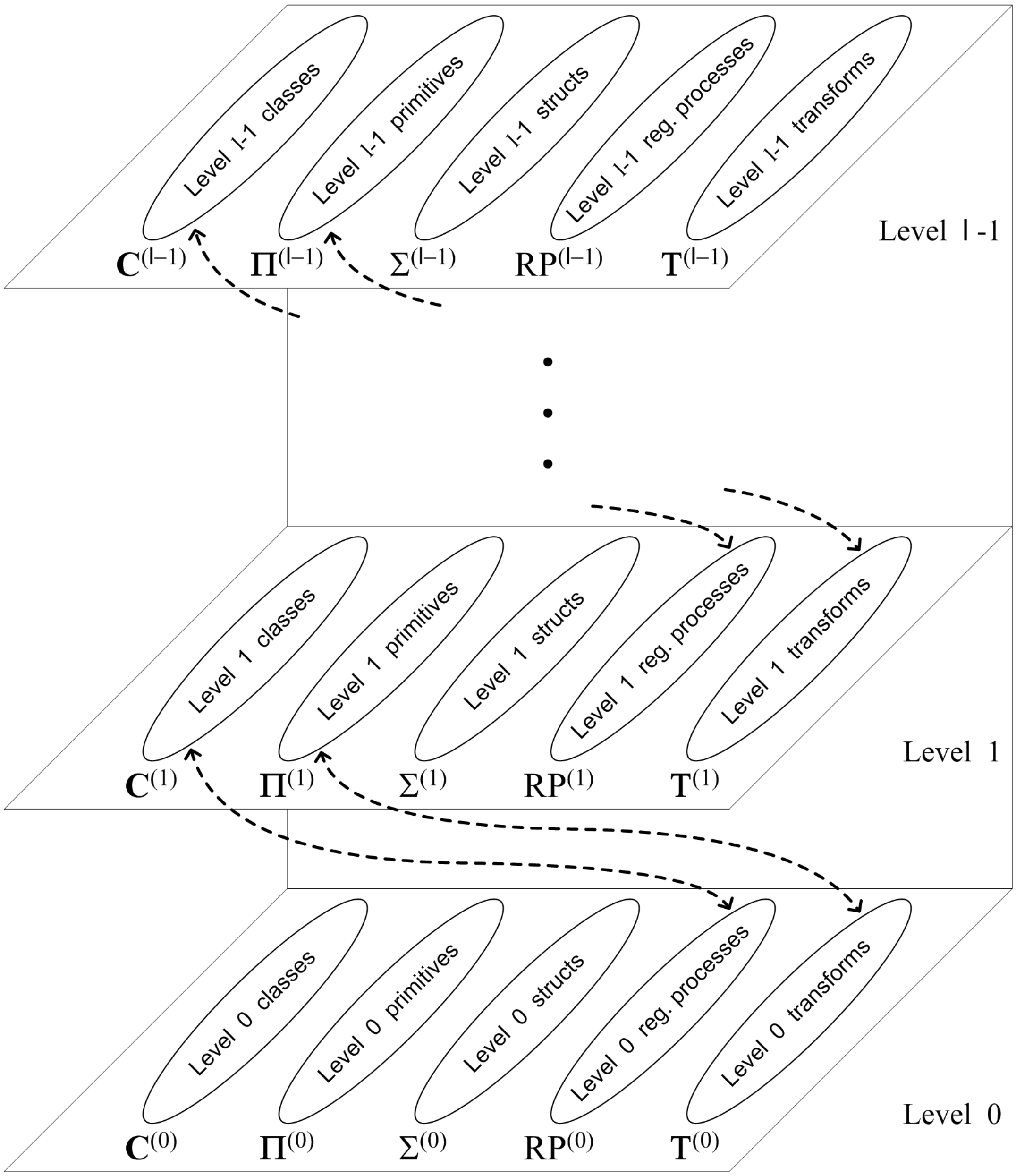

Figure 21: Schematic representation of a multi-level inductive structure with l levels.

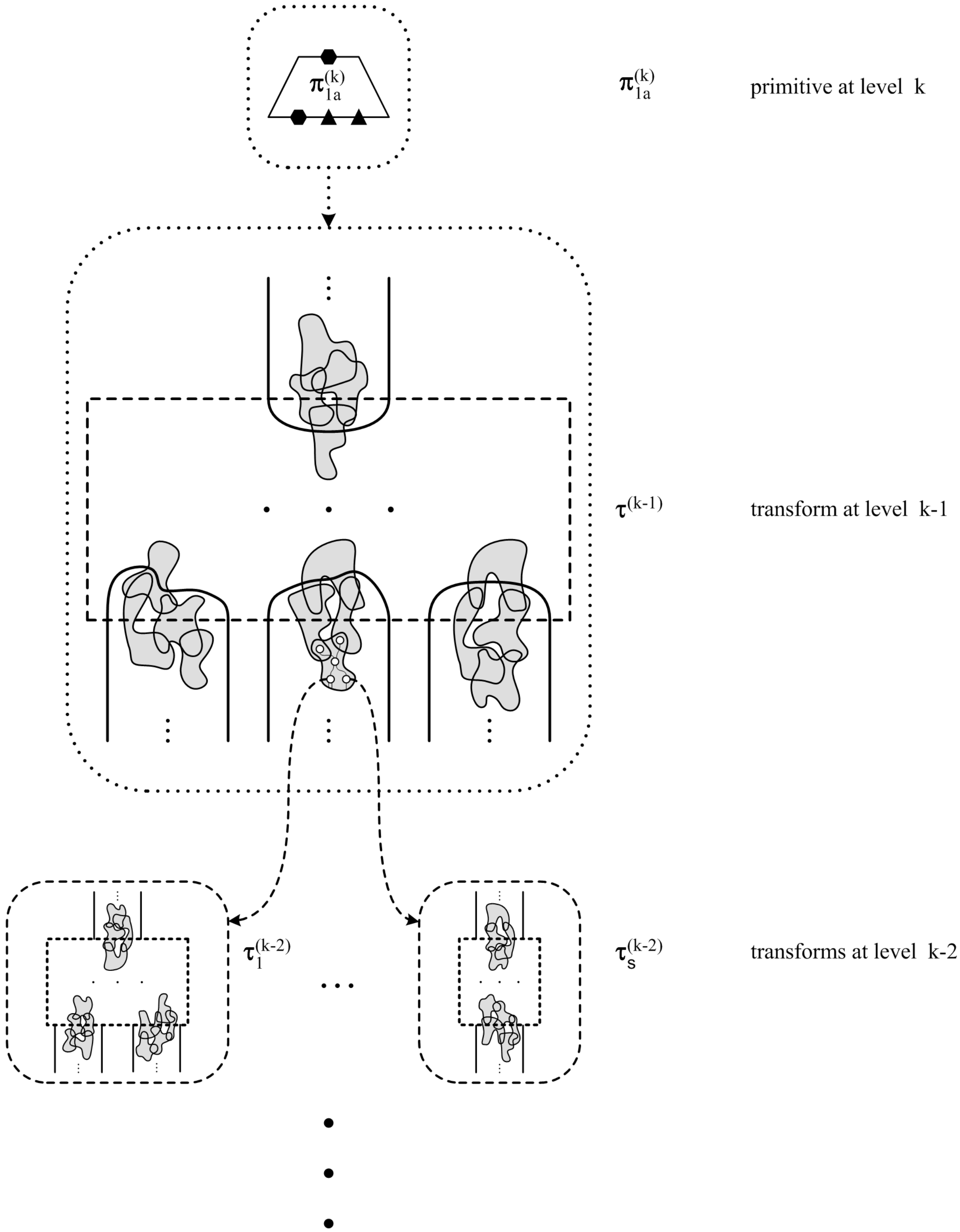

Figure 22: Pyramid view of a k^{th}-level primitive: the pyramid is formed by the corresponding subordinate structural entities (i.e. transforms, generators, primitives, etc.) from previous levels.

Part IV
Conclusion

9 Preliminary thoughts on learning

In this section, we briefly discuss our preliminary ideas regarding (inductive) learning. In general, we strongly believe that, at least for the near future, the emphasis in learning should be less on the statistical activity *called* learning and more on the (structural) representational issues, as suggested, for example, in our class representation postulate in section 6. In our framework, it should not come as a surprise that learning is, in fact, learning of classes and transforms.

As far as the learning of classes is concerned, we mean, of course, the learning of classes of regular structural processes (both initial and terminal). The latter involves the learning of appropriate parameters for the universal class generating system. This will involve, in particular, some optimization process in which these parameters are optimized based on the 'training set'[39].

Since a transform encapsulates the pattern of interaction of several structural processes, having gone through the first cycle of learning of the relevant class representations, one can first identify transform bodies and then modify the initial and terminal class representations for a particular transform if the structure of the this transform's body reveals their inadequacy.

It is also important to remember that at the end of the inductive learning process, in both cases—when learning a new class or updating a previously learned one—the previous *class representation of the object*, if any, (see the Class Representation Postulate on p. 38) is updated to be consistent with the resulting representation of the class.

As far as computational complexity issues are concerned, we note that, *computationally*, learning is bounded by the number of various partitions of the training structs (into generators). The computational cost associated with the construction of a candidate optimal partition appears to be low order polynomial, since, locally, a struct is "almost linear": although the "width" of an entire struct is not constant, the width (and also length) of the generators is always bounded by a small number (since, for all primitives, the 'branching factor' is bounded by the maximal number of terminals). In addition, with each new level of representation, since the size of the next-level struct shrinks by at least an order of magnitude, the complexity of learning at the new level is also reduced. Recall that we have assumed (and this assumption seems to be borne out by reality) that the number of classes at each level is small.

[39] By a training set, we mean a non-conventional analogue of the traditional concept, associated with the result of a sophisticated, dynamic process of interaction between an observer and the target process—mediated by a *structural* (as opposed to numeric) measurement subprocess, which delivers structs.

10 How one should approach the ETS formalism

This section was written to address the situation we often find ourselves in when presenting talks on the ETS formalism. At the end of a talk, instead of discussing the various features of the formalism, we are usually faced with answering "standard" ML/PR questions of the following kind: "Did you compare the performance of your approach with that of neural nets (or some other popular model)?" We would like to stress, here, that this type of question is currently quite unproductive. Such questions presume that an evaluation of ETS should proceed under conventional assumptions, i.e. the *first thing* one should do is to compare the performance of ETS learning algorithms with other learning algorithms. Even a cursory reading of this paper should immediately suggest that the very formulation of the *inductive learning problem* has changed radically and thus "the first thing one should do" is to better understand what this new formulation offers, rather than focusing *prematurely* on non-central issues.

So, which new insights into the nature of the inductive learning problem does the ETS formalism offer? It turns out that the new formulation of this problem (see also a quotation from Vapnik and Chervonenkis given in [66]) can be stated as follows: given a small training set from a class, construct the corresponding *class representation*. As one can now see, for the first time, the concept of class representation comes to the fore. So what is a 'class representation'? Although a hint at the answer to this question was suggested by the concept of generative grammars (i.e. a set of production rules), as the above exposition implies, this answer is inadequate for a more profound reason: the lack of *representational* formalisms in science in general, and in mathematics in particular ([9]). Put simply, there is no adequate formal representational setting for strings as object representations, and therefore all constructions based on such an unsound foundation cannot be effective. Thus, the new problem formulation inevitably focuses ones attention on the concepts of both class and object representation, since the former cannot be properly addressed without the latter. Again, as one can see from this paper, the development of the underlying representational formalism turned out to be an enormous undertaking in which we were faced with unprecedented difficulties.

Returning to the issue of the *evaluation* of the ETS formalism, the immediate focus should be on answering the following kind of scientific questions:

- Is the new (struct) representation more useful and/or powerful as the primary form of representation? That is, given a concrete application, develop an intuition about struct representation in that setting, and then ask the question, "Do these structs contain additional, important information about the actual objects/processes of the application?"[40]

  If the struct concept, as defined in this paper, turns out to be somewhat inadequate as a primary form of representation, can it be repaired within the current overall framework, or does the framework itself have to be rethought?

- Can the proposed class representation postulate fulfill its intended purpose? Is the proposed overall approach to the class generating system a sufficiently powerful way

[40] We believe that there is preliminary support for this claim, e.g. for the first time, an object's formative history is now a part of the representation.

of describing, or representing, actual classes in the chosen applied setting? Is the link between a small training set and its corresponding class representation computationally effective? In fact, it should be clear that the introduced concept of struct substantially reduces the gap between representations of objects and their classes as it exists in the current formalisms.

- What are the options for turning the postulate into a definition? In particular, one should explore empirically, in a concrete application, the nature of the class generating process by building class representations for the corresponding training structs (from primitives, to generators, to "metagenerators", etcetera).

- What are the options for introducing a more adequate concept of transform, aligned with the definition of class generating system (once the above choice of the definition of a class generating system is made)?

References

[1] L. Goldfarb, D. Gay, O. Golubitsky, D. Korkin, What is a structural representation? Second variation, Technical Report TR04-165, Faculty of Computer Science, UNB, 2004.

[2] L. Goldfarb, O. Golubitsky, D. Korkin, What is a structural representation?, Technical Report TR00-137, Faculty of Computer Science, UNB, 2000.

[3] L. Goldfarb, D. Gay, O. Golubitsky, What is a structural representation? Third variation, February 2005 (unpublished).

[4] G. Ifrah, *The Universal History of Numbers*, J. Wiley, New York, 2000.

[5] G. Sarton, *Ancient Science Through the Golden Age of Greece*, Dover, New York, 1993.

[6] E. Schrödinger, *Nature and the Greeks and Science and Humanism*, Cambridge University Press, Cambridge, 1996, pp. 143–145, 158.

[7] L. Goldfarb, Representational formalisms: why we haven't had one, an expanded version of [9], in preparation.

[8] A. W. Crosby, *The Measure of Reality*, Cambridge University Press, 1997.

[9] L. Goldfarb, Representational formalisms: why we haven't had one, *Proc. ICPR 2004 Satellite Workshop on Pattern Representation and the Future of Pattern Recognition*, ed. L. Goldfarb, Cambridge, UK, August 2004.

[10] L. Goldfarb, On the foundations of intelligent processes I: An evolving model for pattern learning, *Pattern Recognition* 23 (6), 1990, pp. 595–616.

[11] L. Goldfarb, What is distance and why do we need the metric model for pattern learning, *Pattern Recognition* 25 (4), 1992, pp. 431–438.

[12] L. Goldfarb, S. Nigam, The unified learning paradigm: A foundation for AI, in: V. Honavar, L. Uhr (eds.), *Artificial Intelligence and Neural Networks: Steps toward Principled Integration*, Academic Press, Boston, 1994, pp. 533–559.

[13] L. Goldfarb, J. Abela, V. C. Bhavsar, V. N. Kamat, Can a vector space based learning model discover inductive class generalization in a symbolic environment?, *Pattern Recognition Letters* 16 (7), 1995, pp. 719–726.

[14] L. Goldfarb, Inductive class representation and its central role in pattern recognition, *Proc. Conf. Intelligent Systems: A Semiotic Perspective*, Vol. 1, NIST, Gaithersburg, Maryland, USA, 1996, pp. 53–58.

[15] L. Goldfarb, What is inductive learning? Construction of inductive class representation, *Proc. Workshop "What Is Inductive Learning" in Conjunction with 11th Biennial Canadian AI Conf.*, 1996, pp. 9–21.

[16] L. Goldfarb, S. Deshpande, What is a symbolic measurement process?, *Proc. IEEE Conf. Systems, Man, and Cybernetics*, Vol. 5, Orlando, Florida, USA, 1997, pp. 4139–4145.

[17] L. Goldfarb, J. Hook, Why classical models for pattern recognition are not pattern recognition models, *Proc. Intern. Conf. On Advances in Pattern Recognition*, Plymouth, UK, 1998, pp. 405–414.

[18] K. S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[19] M.A. Aiserman, Remarks on two problems connected with pattern recognition, in: S. Watanabe (ed.), *Methodologies of Pattern Recognition*, Academic Press, 1969, p. 1.

[20] S. Wermter, R. Sun, *Hybrid Neural Systems*, Springer-Verlag, Heidelberg, 2000.

[21] H. Bunke, A. Kandel (eds.), *Hybrid Methods in Pattern Recognition*, World Scientific, 2002.

[22] 3rd International Workshop on Hybrid Methods for Adaptive Systems, Oulu, Finland, July 2003.
http://adiret.cs.uni-magdeburg.de/∼nuernb/hmas2003/

[23] 4th International Workshop on Hybrid Methods for Adaptive Systems, Aachen, Germany, June 2004.
http://adiret.cs.uni-magdeburg.de/hmas2004/

[24] 2004 AAAI Fall Symposium Series Workshop on Compositional Connectionism in Cognitive Science, Washington, D.C., October 2004.
http://www.cs.wlu.edu/∼levy/aaai04/

[25] SRL2004 Workshop: Statistical Relational Learning and its Connections to Other Fields, Banff, Canada, July 2004.
http://www.cs.umd.edu/projects/srl2004/

[26] MRDM 2005: 4th Workshop on Multi-Relational Data Mining, Chicago, August 2005.
http://www-ai.ijs.si/SasoDzeroski/MRDM2005/

[27] Machine Learning Journal Special Issue on Multi-relational Data Mining and Statistical Relational Learning, 2005.
http://www.cs.kuleuven.ac.be/∼ml/mrdm-srl.html

[28] A. Bird, *Philosophy of Science*, McGill-Queen's University Press, Montreal, 1998.

[29] J. H. Holland, K. J. Holyoak, R. E. Nisbett, P. R. Thagard, *Induction*, MIT Press, Cambridge, Mass., 1986.

[30] J. Losee, *A Historical Introduction to the Philosophy of Science*, 3rd ed., Oxford University Press, Oxford, 1993.

[31] H. Margolis, *Patterns, Thinking, and Cognition*, University of Chicago Press, 1987, pp. 1, 3.

[32] E. G. H. Landau, *Foundations of Analysis*, Chelsea, New York, 1951.

[33] C. Lee, *Notes for Math 502*, 1998.
http://www.ms.uky.edu/∼lee/ma502/notes2/node7.html

[34] N. Chomsky, *Knowledge of Language: Its Nature, Origin, and Use*, Praeger, New York, 1986, p. 12.

[35] M. Piattelli-Palmarini (ed.), *Language and Learning: The Debate between Jean Piaget and Noam Chomsky*, HUP, Cambridge, USA, 1980, pp. 100–103, 255–272.

[36] M. Leyton, *Symmetry, Causality, Mind*, MIT Press, Cambridge, Mass., 1992, p. 1–2.

[37] A. R. Lacey, *A Dictionary of Philosophy*, 3rd ed., Routledge, London, UK, 1996, p. 308.

[38] R. Dunbar, *The Trouble with Science*, Faber and Faber, London, UK, 1996, p. 17.

[39] O. Golubitsky, *On the generating process and the class typicality measure*, Technical Report TR02-151, Faculty of Computer Science, UNB, 2002.

[40] O. Golubitsky, *On the Formalization of the Evolving Transformation System Model*, Ph.D. thesis, Faculty of Computer Science, UNB, March 2004.

[41] D. Korkin, *A New Model for Molecular Representation and Classification: Formal Approach Based on the ETS Framework*, Ph.D. thesis, Faculty of Computer Science, UNB, 2003.

[42] D. Clement, *Information Retrieval via the ETS Model*, Master's thesis, Faculty of Computer Science, UNB, 2003.

[43] S. Falconer, D. Gay, L. Goldfarb, ETS representation of fairy tales, *Proc. ICPR 2004 Satellite Workshop on Pattern Representation and the Future of Pattern Recognition*, ed. L. Goldfarb, Cambridge, UK, August 2004.

[44] S. Falconer, *On the Evolving Transformation System Model Representation of Fairy Tales*, Master's thesis, Faculty of Computer Science, UNB, 2005.

[45] M. Al-Digeil, *Towards an Evolving Transformation System Representation of Proteins*, Master's thesis, Faculty of Computer Science, UNB, 2005 (in progress).

[46] A. Gutkin, *Towards Formal Structural Representation of Spoken Language: An Evolving Transformation System (ETS) Approach*, Ph.D. thesis, School of Informatics, University of Edinburgh, 2005 (submitted).

[47] J. M. Abela, *ETS Learning of Kernel Languages*, Ph.D. thesis, Faculty of Computer Science, UNB, 2002.

[48] V. N. Kamat, *Inductive Learning with the Evolving Tree Transformation System*, Ph.D. thesis, Faculty of Computer Science, UNB, 1995.

[49] S. Nigam, *Metric Model Based Generalization and Generalization Capabilities of Connectionist Models*, Master's thesis, Faculty of Computer Science, UNB, 1993.

[50] B. K. Hall and W. M. Olson (eds.), *Keywords and Concepts in Evolutionary Developmental Biology*, Harvard University Press, Cambridge, USA, 2003.

[51] S. B. Carol, *Endless Forms Most Beautiful: The New Science of Evo Devo*, Norton, New York, 2005.

[52] S. B. Carol, J. K. Grenier, S. D. Weatherbee, *From DNA to Diversity*, 2nd ed., Blackwell, Massachusetts, 2004.

[53] L. Wolpert, *Triumph of the Embryo*, Oxford University Press, Oxford, 1992.

[54] R. Schlegel, *Time and the Physical World*, Dover, New York, 1968.

[55] W. H. Cropper, *Great Physicists*, Oxford University Press, Oxford, 2001, p. 277.

[56] J. Jeans, *The New Background of Science*, University of Michigan Press, Ann Arbor, 1959, pp. 293, 295.

[57] H. A. Simon, *The Sciences of the Artificial*, MIT Press, Cambridge, Mass., 1996.

[58] R. P. Feynman, *QED: The Strange Theory of Light and Matter*, Penguin Books, London, UK, 1990.

[59] G. Kane, *The Particle Garden: Our Universe as Understood by Particle Physicists*, Addison-Wesley, Reading, Massachusetts, 1995, Appendix A.

[60] T. P. Smith, *Hidden Worlds: Hunting for Quarks in Ordinary Matter*, Princeton University Press, Princeton, 2003.

[61] H. Quinn, *Theory: Feynman Diagrams*, 2003.
http://www2.slac.stanford.edu/vvc/theory/feynman.html

[62] N. David Mermin, *Boojums All the Way through*, Cambridge University Press, Cambridge, 1990.

[63] Amir Aczel, *Entanglement*, Plume, New York, 2003.

[64] L. Wolpert et al., *Principles of Development*, 2nd ed., Oxford University Press, Oxford, 2002.

[65] S. F. Gilbert, *Developmental Biology*, 7th ed., Sinauer, Sunderland, Massachusetts, 2003.

[66] ICPR 2004 Satellite Workshop on Pattern Representation and the Future of Pattern Recognition: A Program for Action, chair L. Goldfarb, Cambridge, UK, August 2004.
http://www.cs.unb.ca/~goldfarb/conf/ICPR-2004_Workshop.html