What is a structural representation?

Second version*

Lev Goldfarb, David Gay, Oleg Golubitsky, Dmitry Korkin

Faculty of Computer Science
University of New Brunswick
Fredericton, Canada

April 3, 2004

Abstract

We outline a formalism for "structural", or "symbolic", representation, the necessity of which is acutely felt in all sciences. One can develop an initial intuitive understanding of the proposed representation by simply generalizing the process of construction of natural numbers: replace the identical structureless units out of which numbers are built by several structural ones, attached consecutively. Now, however, the resulting constructions embody the corresponding formative/generative histories, since we can see *what* was attached and *when*.

The concept of class representation—which inspired and directed the development of this formalism—differs radically from the known concepts of class. Indeed, the evolving transformation system (ETS) formalism proposed here is the first one developed to support that concept; a class representation is a finite set of weighted and interrelated transformations ("structural segments"), out of which class elements are built.

The formalism allows for a *very natural* introduction of representational levels: a next-level unit corresponds to a class representation at the previous level.

We introduce the concept of "intelligent process", which provides a suitable scientific environment for the investigation of structural representation. This process is responsible for the actual construction of levels and of representations at those levels; conventional "learning" and "recognition" processes are integrated into this process, which operates in an unsupervised mode. Together with the concept of structural representation, this leads to the delineation of a new field—inductive informatics—which is intended as a rival to conventional information processing paradigms.

From the point of view of the ETS formalism, classical discrete "representations" (strings, graphs) now appear as incomplete special cases at best, the proper "completion" of which should incorporate corresponding generative histories (e.g. those of strings or graphs).

* Technical Report TR04-165, Faculty of Computer Science, UNB. This paper is a substantially modified version of [19].

> [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

Part I
Prolegomenon

> [W]e are all waiting, not necessarily for a recipe, but for new techniques for apprehending the utterly remote past. Without such a breakthrough, we can continue to reason, speculate, and argue, but we cannot know. Unless we acquire novel and powerful methods of *historical* enquiry, science will effectively have reached a limit. [Emphasis added.]
>
> F. M. Harold, The Way of the Cell, 2001

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.

On the one hand, although the overwhelming importance of structural/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]. (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".) On the other hand, 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 both 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 of the above obstacles is usually underestimated or overlooked entirely.

[1] The Chomsky formal grammar model will be discussed later in the Introduction. We are not aware of any other (not *derived* from the Chomsky approach) *basic* attempts at structural/symbolic representation.

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 [23], [35]. 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].

The overwhelming preponderance 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 complexity of the task.

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 [11]–[18].

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/structural approach [9], each possessing the desirable features lacking in the other—should be unified [2]:

> 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 naive, not so much with respect to timeliness but with

[2] There are, of course, rare exceptions (see [36], for example).

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

[4] For an insightful explanation of how the stage was set for this, see [7].

respect to the (underestimated) novelty of such a unified formalism: there was no formal framework which could naturally accommodate unification [11]. It is interesting to note that researchers working in the various "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. [4], [38], [40], [41]) attests to the rediscovery of the need for unification.

1.3 The general approach 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 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 modelling 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 objects? In other words, how do we formally capture the more general inductive (or generative) process of object construction? What is the connection between a class description/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 and how do they communicate during the intelligent process? It is understood that all of the above must be accomplished *naturally* within a single general model.

On the formal side, we chose the path of a far-reaching generalization of the Peano axiomatization/construction of natural numbers ([28] or [29]), 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 identical structureless primitives out of which natural numbers are built (Fig. 8, p. 19) by various structural ones (Fig. 7, p. 18). Then, one can build on that foundation.

[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, [3], [22], [31]). 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" [32].

1.4 ETS as the first representational formalism

We now strongly believe that the concept of structural object representation cannot be divorced from that of "evolutionary" object representation, i.e. a representation capturing a generative history of the object. Herein, we believe, lies the fundamental difference between classical numeric and "structural/symbolic" representations. In light of this, widely-used nonnumeric "structural representations" such as strings, trees, and graphs cannot, in our opinion, be considered as such. In short, since such "representations" do not encode the generative object history[6], there is very little connection between the corresponding object and the class of objects with respect to which (in the current context) the object is to be represented. Moreover, 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 [5], [34]). In particular, the generative issues so important to Chomsky cannot be properly addressed within the "string" setting, mainly because a string *cannot* capture the object's formative history; there are exponentially many formative histories[7] hidden behind a string.

With respect to "evolutionary" 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 [30]:

> [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.

The concept of a "representational formalism" will be discussed in [20]. Here we simply mention that according to the view expressed therein, we presently have only one, "primitive", representational formalism (excluding the ETS model), 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.

What would be an appropriate scientific environment for understanding and investigating the nature of structural representation? It appears that such an environment is provided by the concept of the intelligent process, also introduced in this paper. This process is

[6] 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.

responsible for the actual construction of (representational) levels and of representations at each of those levels. Moreover, it integrates learning and recognition processes and operates in an unsupervised mode, to use a standard term from PR. As mentioned in the abstract, we expect this scientific environment, together with the concept of structural representation and its various application areas (e.g. pattern recognition, data mining, information retrieval, bioinformatics, molecular phylogenetics, cheminformatics) to delineate a new information processing paradigm: inductive informatics.

In light of the obvious monumental difficulties related to the development of a formal model for structural representation, the best we can hope for as a result of the present attempt is to propose and outline the 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. 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 interpreted.

Ultimately, what should make or break the ETS model as a representational model? 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 do not insist on any form of (inductive) class representation, but simply *adapt* existing formalisms to fit the learning problem (without the availability of an adequate concept of inductive class representation in such formalisms). Thus, an immediate value 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 as philosophical deliberation.

The model's basic tenets both "explain" the nature of the inductive learning process 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 ..." [27] and "Whatever else science is used for, it is explanation that remains its central aim" [8]. Current inductive learning models explain essentially nothing about the nature of this, quite possibly central, intelligent process.

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 "ideology" of the framework.

Finally, a peculiar view of the resulting formalism is that of a multi-level "representational chemistry". In fact, the ETS model suggests a very different picture of reality than that implied by modern mathematics: equational descriptions of physical reality are replaced by structural descriptions of evolving classes of objects in the universe. We believe that the proposed formalism provides radically new insight into the nature of things and offers a guiding metaphor badly needed by various sciences (see, for example, the epigraphs to Parts I, II). As to progress in the development 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, we presently lack.

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 "intelligent process"[8], and Part II presents the main formal concepts (sections 3–7). We must admit that, as far as a final/satisfactory (applied) interpretation of a primitive transformation is concerned, we are not there yet. In Part III we give a provisional sketch of the intelligent process (sections 8–13). Part IV lists high-priority directions and some last-minute ideas (sections 14,15). Section 15 replaces a stronger requirement of linear ordering of primitive transformations by their partial ordering, and was added as an afterthought to help other researchers (who may want to proceed with applications of the model) to more appropriately adapt the main concepts to the needs of applications. The material presented in Part III is in a considerably more tentative state than that of Part II, and will be continuously revised as our experience with concrete data accumulates. We have undertaken this very tentative description of the process in order to give the reader some *process* view, as opposed to a "static" view, of the ETS model.

Although we warn the reader about this situation in section 3 (see the reading suggestion on p. 21), we must also mention the issue here: the exposition was *substantially* burdened by the unfortunate technical necessity of carrying labels to identify as well as to distinguish (intrinsic) "sites" in various structural entities. In the long term, we expect this situation to change substantially as the development of the model progresses. It is also useful to keep in mind that the *interplay between structural and numeric aspects* in the current version of the intelligent process has not yet reached a satisfactory state, although it should be quite clear that structural considerations already transcend numeric ones. A more satisfactory state would entail a much more seamless subjugation of numeric aspects to structural ones. However, despite these and various other shortcomings of the present version of the formalism mentioned throughout the paper, we believe that its mastery may be quite useful before a transition to the next version is made, both from theoretical and applied considerations.

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 larger than we anticipated. Moreover, the relatively large number of definitions is easily explained by the absolute novelty of the formalism and of all the basic concepts. (Obviously, these features increase the conceptual density of the paper even further.) We warn the reader that the technical difficulty of the paper gradually increases, culminating in section 13, which is why we added an index of the main concepts as an appendix.

[8] This term will be immediately clarified in section 2.

For an alternative exposition of the ETS model which outlines the first formalization of levels (different from the present), see [21]. Some very preliminary applications to cheminformatics are discussed in [25] and to information retrieval in [6]. Three theses presenting some earlier "pre-formal" work on the ETS learning algorithms are [1], [24], and [33].

We thank Alexander Gutkin and Muhammad Al-Digeil for their help with proofreading.

2 The ETS tenet: the evolution of the universe as the evolution of the intelligent process

The ETS framework is inspired by the view of the universe as the evolving intelligent process, spawning other intelligent subprocesses. What do we mean by an intelligent process?

By an intelligent process[9], we understand an actual non-deterministic process operating on structured actual entities by assembling them into larger entities, guided by some "abstract description". The latter does not mean that the process "reads" this description, but rather that its instantiation is guided in some particular manner. The spawning of a new subprocess is related to the "recognition" of some recurring parts of the current process as modular units/subprocesses. This recognition, in turn, modifies their instantiations as well as the corresponding descriptions. In general, the ETS model suggests 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 modular units (which will be called class supertransformations below) are instantiated. In other words, once the process begins to assemble larger (longer-running) subprocesses, the overall generation time increases. Historically, some of these transitions were associated with transitions to new levels: e.g. atomic levels, molecular levels, etc. In the ETS model, a transition to a new level is related to the declaration of a newly selected class supertransformation as a primitive transformation, where the latter is the postulated *primitive*, or atomic, building block (see Fig. 1).

Moreover, it is important to emphasize that the concept of an intelligent process now becomes more fundamental than that of class, i.e. class objects become a product of the intelligent process, where each object is an epiphenomenon resulting from a (local) self-assemblage of faster-running intelligent subprocesses (or process parts). For example, a single (vibrating) water molecule, observed over a particular interval, should be thought of as being *continuously regenerated* by a "molecular" intelligent process acting on faster-running "atomic" intelligent processes. Also, the regeneration of *different* water molecules is guided by the *same* description, and thus one can speak of a class of water molecules which is a manifestation of this intelligent process. In science, the corresponding intelligent process has, so far, remained behind the scene, while its products are more familiar to us.

2.1 Transformations, classes, and intelligent processes

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

[9] For a formal definition, see Part III.

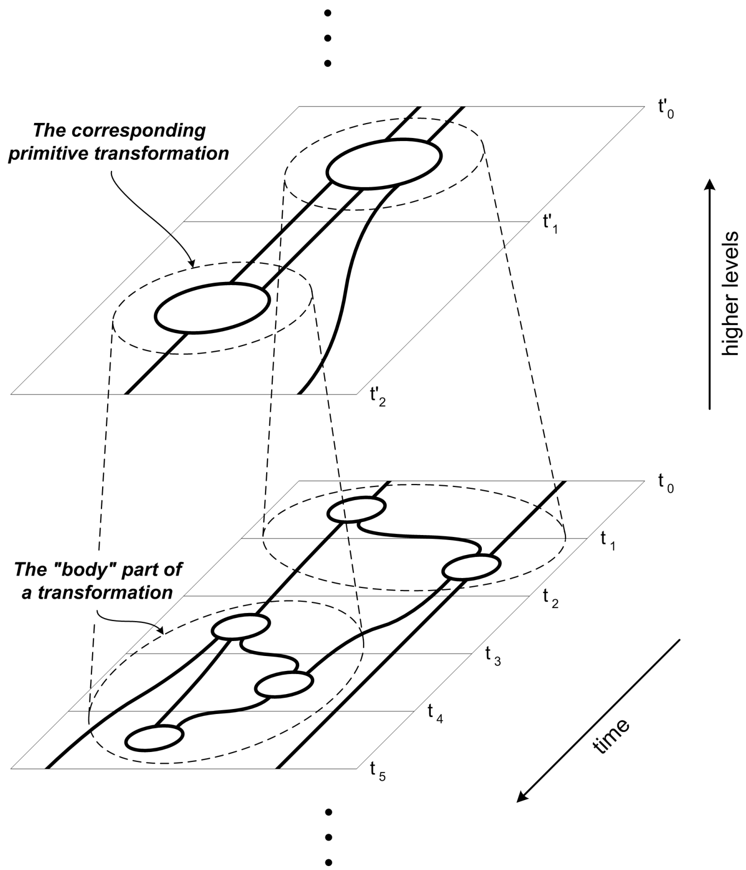


Figure 1: Simplified multi-level ETS representation with different time scales for each level. The circles denote "events" and the lines between them represent their "shared attributes". (Two *consecutive* levels are shown. The time scale for the higher level is measured in coarser units, i.e. t'_0 corresponds to t_0, t'_1 corresponds to t_2, etc.) The shown supertransforms consist of single transformations, and the "context" parts of the transformations are not identified.

ubiquity of the class concept, many areas of information processing—e.g. pattern recognition (including speech and image recognition), machine learning, neural networks, data mining, information retrieval, bioinformatics, cheminformatics—rely on this concept as the central one. Since these areas have to deal effectively with this concept, of necessity they must rely on some formalisms. However, as was mentioned in the fourth last paragraph of section 1.4, conventional formalisms were not developed to address the needs of class representation, so one has to adapt them for this purpose; these adaptations become an obstacle to be overcome. Note that even natural languages simply *name* classes of objects/events rather than capture their (evolving) *representations*.

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 set of structural transformations; these transformations are supposed to play the role of structural units, out of which class objects are assembled. Accordingly, within the ETS model, a class description is formally encapsulated by the concept of "supertransformation" (Def. 16), which is a set of interrelated transformations.

The concept of an intelligent process bridges the gap between the (abstract) class description and the actual class objects. Moreover, because this concept breathes life into the concept of class (by encapsulating the manner in which various class objects are formed), it should be viewed as the central concept in the ETS framework. In general, the intelligent process orchestrates the manner in which class transformations interact during the various subprocesses of object formation. The object's *formative/generative history* can thus be easily inferred by observing the workings of the corresponding intelligent subprocess. By an object's generative (or formative) history, we mean the sequence of transformations in the intelligent process involved in the generation/construction of the object (see also 2.3).

2.2 Objects as epiphenomena of intelligent processes

As the section heading implies, according to the ETS model, visible objects are not what they appear to be. This point is not as controversial as it seems if all objects were treated as organisms, having developmental as well as evolutionary histories. Thus, a developed organism should, more accurately, be viewed as an epiphenomenon of both these histories; 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. In fact, when we look at such an object as a chair, it also has its "developmental" (i.e. production) and "evolutionary" (i.e. conceptual) histories. The ETS model suggests such a view of reality.

It appears one can "blame" 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, 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 levelwise, which is what we observe in the universe. For example, molecular structures are now inconceivable without reference to atomic structures. It appears to us (as well as to other scientists and philosophers, e.g. Sec. 1.4) that the invisible processes responsible for the generation of visible structure are of primary interest, as compared to the visible structures themselves.

2.3 Emergence of levels in an intelligent process

How should the evolution of an intelligent process be viewed? Since every intelligent process has two main aspects, generative and interactive (with other processes), its evolution should be viewed in light of these aspects. The generative aspect is related to the ability of the process to generate various objects of previously-discovered classes. And, of course, every intelligent process regularly exercises this ability: e.g., as was mentioned at the beginning of Section 2, a molecular intelligent process generates concrete class elements (molecules). The interactive aspect of an intelligent process refers to its interaction with other processes. New subprocesses are spawned as a result of such interactions: e.g. the formation of the H_2O intelligent process is the result of the interactions of the H and O processes. Of course, all such interactions occur within the purview of the universal intelligent process.

How do levels (of representation) emerge in the ETS framework? In short, the intelligent process, described in Part III, creates them as it evolves. The emergence of each new level corresponds to the discovery by the process of the first "supertransform"[10] (loosely, a collection of related transformations) at the currently highest level. Together with the new level, a new primitive transformation is created on the basis of that supertransform. Discovery by the process of subsequent supertransforms at any level expands the set of primitive transformations at the next level. The input for the above process is a sequence of "sensory events", captured as primitive transformations of the initial level (see Fig. 2)

A useful metaphor for capturing the structure of an intelligent process is a multi-level "evolving representational tower" which can "see" (and interact with) data *only* at the initial level. In the language of the ETS formalism, each level records/represents data by means of its own "primitives", with which it replaces a structural fragment from a previous level (thus changing the *representation* of the data). Each level k of this tower is responsible for the detection of regularities in the data at the k-th resolution level, relying on the (condensed) data representation passed up from the previous level (see Fig. 2).

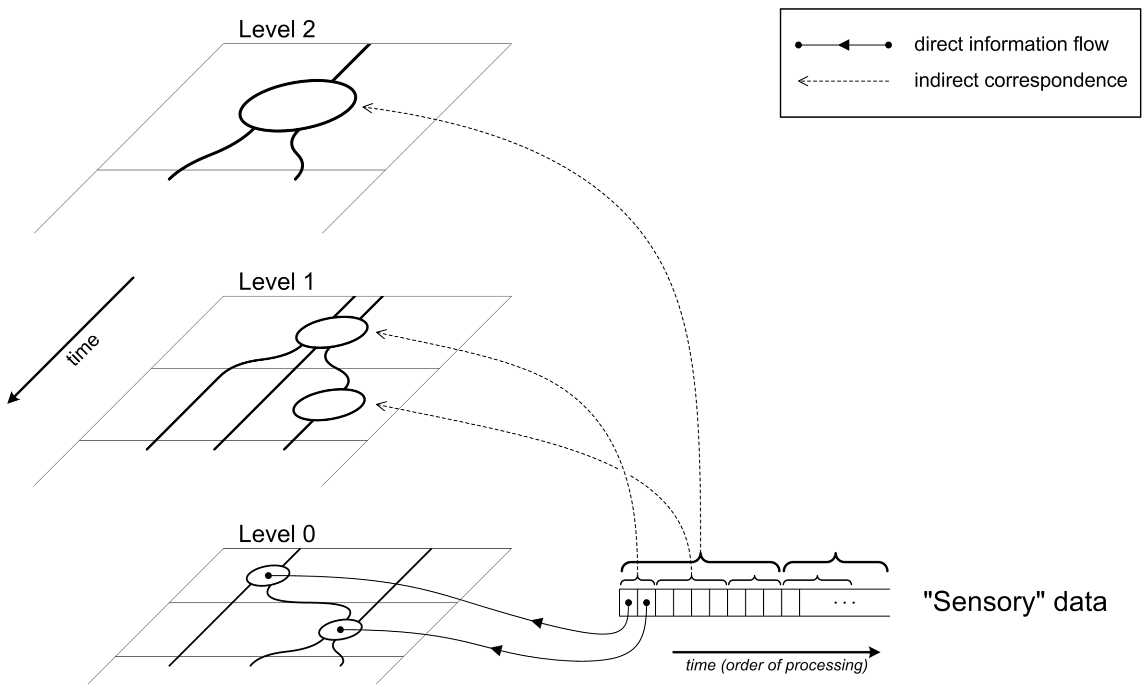


Figure 2: A multi-level representational tower with a single-level sensor at level 0.

2.4 Working assumptions

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

[10] See Def. 16.

object view encapsulates the *common* scientific view of reality, while the proposed event view encapsulates the ETS, or intelligent 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/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 *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). As mentioned above, the latter view of the environment, which we call the *event environment*, insists on the primacy of the intelligent process rather than on the primacy of the objects themselves.

Given the present state of science, i.e. an object-centered view, we have decided 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. 3). 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.

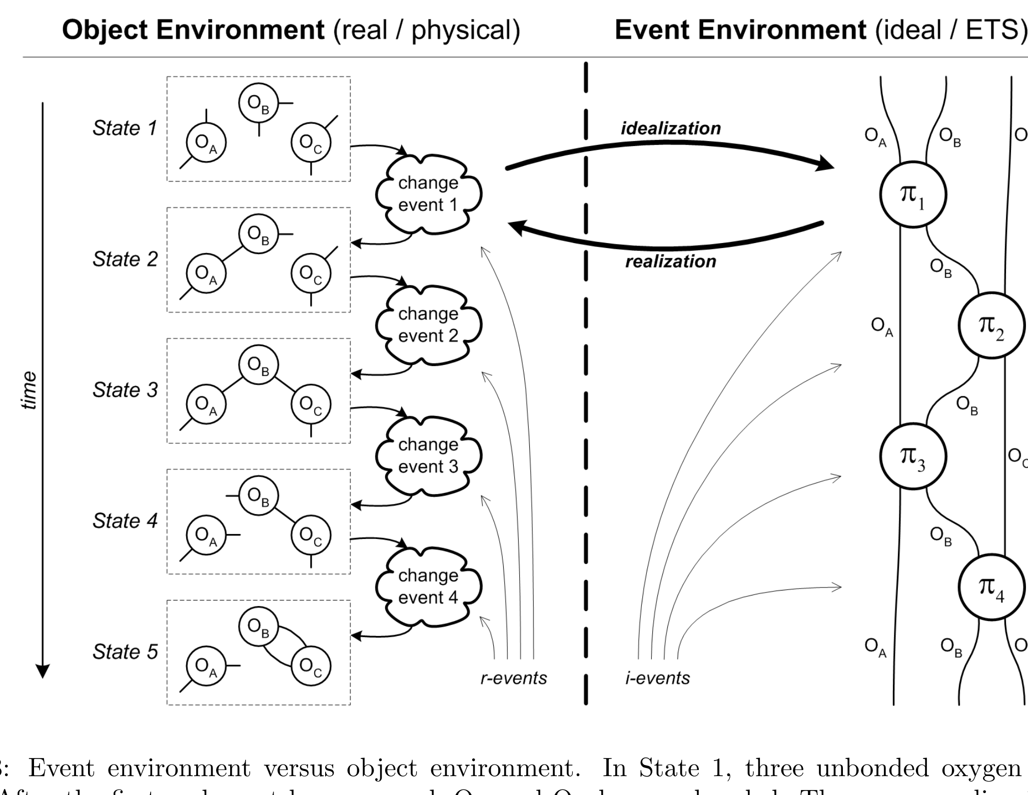


Figure 3: Event environment versus object environment. In State 1, three unbonded oxygen atoms are shown. After the first real event has occurred, O_A and O_B become bonded. The corresponding ideal event (primitive π_1) is depicted on the right. Three subsequent state changes are also depicted.

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

> The structure of organisms has been studied with great intensity without corresponding advances in the fundamental theory of organization.
> ……………………………………………………………………
> What is missing? Possibly knowledge of certain crucial facts, but certainly a sufficiently clear formulation of the problem. We cannot expect to understand organization until we know what we are looking for *in terms of mathematics*. The use of mathematical standards alone can clarify the aim of a theory of organization.
>
> L. L. Whyte, Internal Factors in Evolution, 1965

3 Primitive transformations and class primitives

In this section we introduce the basic constructive elements of the model, in our case the elementary transformations or *"elementary ideal events"*. Note that the ETS model suggests that transformations appear in the universe before objects, consistent with the primacy of the process view of reality.

We 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.

Definition 1. Let

$\widehat{\Pi} = \{\widehat{\pi}_1, \widehat{\pi}_2, \cdots, \widehat{\pi}_n\}$	be the set of **names of primitives** (small finite set),
SL	be the set of **site labels** (finite or countable set),
ST	be the set of **site types** (small finite set).

Moreover, $\forall\, \widehat{\pi}_i \in \widehat{\Pi}$, we are given a triple

$$\mathring{\pi}_i = \langle \widehat{\pi}_i, \mathrm{INIT}_i, \mathrm{TERM}_i \rangle$$

called an **original primitive transformation**, or simply **original primitive**, where $\mathrm{INIT}_i, \mathrm{TERM}_i$ are (small) finite, possibly empty, *linearly ordered* sets of site labels, of cardinalities k, l, respectively, $k + l \neq 0$, $]\mathrm{INIT}_i[\subseteq SL$, $]\mathrm{TERM}_i[\subseteq SL$.[11] We denote by $\mathring{\Pi}$ the finite set comprised of $\mathring{\pi}_i$, $1 \leq i \leq n$, and call it the **set of original primitives**. Finally, we are also given a **site type mapping**

$$\mathrm{TYPE} : SL \to ST.$$

►

The above mapping TYPE simply assigns a site type to each site label. To simplify the following terminology, we will use the terms "site" and "site label" interchangeably, although one should keep in mind that site labels are the more auxiliary concept in the sense that, as the name suggests, site labels are used only for *labeling* sites.

Definition 2. For an original primitive $\mathring{\pi}_i$, we introduce the following concepts and notations:

$$\begin{array}{rcll}
\mathrm{Init}\,(\mathring{\pi}_i) & \stackrel{\text{def}}{=} &]\mathrm{INIT}_i[& \text{is the set of } \textbf{initial sites} \text{ of } \mathring{\pi}_i \\
\mathrm{Term}\,(\mathring{\pi}_i) & \stackrel{\text{def}}{=} &]\mathrm{TERM}_i[& \text{is the set of } \textbf{terminal sites} \text{ of } \mathring{\pi}_i \\
\mathrm{Sites}\,(\mathring{\pi}_i) & \stackrel{\text{def}}{=} &]\mathrm{INIT}_i[\;\cup\;]\mathrm{TERM}_i[& \text{is the set of } \textbf{all sites} \text{ of } \mathring{\pi}_i \\
\mathring{\pi}_i(\overline{k}) & & & \text{is the } k\textbf{-th initial site} \text{ of } \mathring{\pi}_i \\
\mathring{\pi}_i(\underline{l}) & & & \text{is the } l\textbf{-th terminal site} \text{ of } \mathring{\pi}_i.
\end{array}$$

►

Remark 1. Although, as was mentioned in section 1.5, we cannot present a "final" (applied) interpretation of a primitive transformation, we may suggest the following tentative interpretation: original primitives are basic, or initial, transformations of input processes into output ones, i.e. a primitive captures the result of the "standard" interaction of input processes. From a conventional PR point of view, primitives are "features" and their sites are the feature's contexts, i.e. sites specify how features are "embedded" in an object. Moreover, we note that a site type encapsulates the inherent structural or qualitative character of a site, while site labels are merely temporary, interchangeable names (see Def. 4); a site type specifies the kinds of allowable "interactions" of this site with the sites of other primitives.

Pictorially, it is convenient to represent an original primitive $\mathring{\pi}_i = \langle \widehat{\pi}_i, \mathrm{INIT}_i, \mathrm{TERM}_i \rangle$ as a convex shape[12]. The initial sites are marked as points on its top, and the terminal sites

[11] For a linearly ordered set $\mathscr{A} = \langle A, < \rangle$, $]\mathscr{A}[\stackrel{\text{def}}{=} A$, i.e. $]\mathscr{A}[$ is the set obtained by discarding the linear order in $\mathscr{A}$. In our case, the linear order is necessary for technical, rather than any intrinsic, reasons.

[12] In a concrete application the use of different shapes helps one to distinguish between various classes of primitives (e.g. Fig. 5).

are marked on its bottom. We will use numbers as labels for the sites, with the left-to-right ordering of the sites on the top and bottom corresponding to the linear orderings in INIT_i and TERM_i, respectively (Fig. 4). Note that natural numbers are used as labels incidentally, and only for convenience.

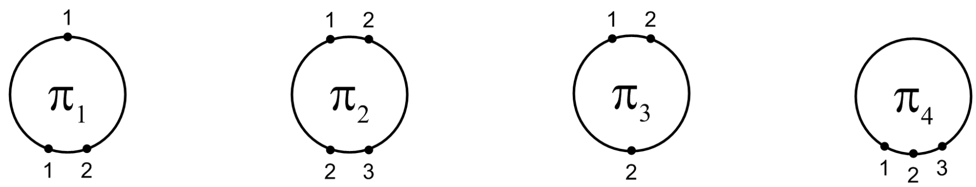


Figure 4: Pictorial illustration of four original primitives, $\mathring{\pi}_1 = \big\langle \widehat{\pi}_1, \langle 1\rangle, \langle 1,2\rangle \big\rangle$, $\mathring{\pi}_2 = \big\langle \widehat{\pi}_2, \langle 1,2\rangle, \langle 2,3\rangle \big\rangle$, etc (no relation to those shown in Fig. 3). Note that the $\widehat{\ }$ symbols are dropped in this and *all* subsequent figures, and site types are not indicated.

The following definition will be useful throughout the paper.

Definition 3. A **site relabeling** F is an injective mapping,

$$F : L \to SL$$

where $L \subset SL$, preserving site types, i.e.

$$\forall\, x \in L \qquad \mathrm{TYPE}\,(x) = \mathrm{TYPE}\,\big(F(x)\big).$$

►

The next definition introduces the concept of site relabeling for primitives, which is necessary for the detection of *structurally identical* primitives. As always, for a mapping $f : X \to Y$ and $A \subseteq X$, $f\big|_A$ is the *restriction of* f *to* A, and the notation id stands for the identity mapping on the appropriate set.

Definition 4. For an original primitive $\mathring{\pi}_i = \langle \widehat{\pi}_i, \mathrm{INIT}_i, \mathrm{TERM}_i \rangle$, a site relabeling f,

$$f : \mathrm{Sites}\,(\mathring{\pi}_i) \to SL,$$

is called a **site relabeling of original primitive** $\mathring{\pi}_i$. Moreover, a (non-original) **primitive transformation**, or simply **primitive**, denoted $\mathring{\pi}_{i\{f\}}$, is defined as

$$\mathring{\pi}_{i\{f\}} \stackrel{\text{def}}{=} \langle \widehat{\pi}_i, f(\mathrm{INIT}_i), f(\mathrm{TERM}_i) \rangle,$$

where the linear orders on $f(\mathrm{INIT}_i)$ and $f(\mathrm{TERM}_i)$ are induced by those in INIT_i and TERM_i, respectively.

Correspondingly, the set Π_i of **structurally identical primitives** is defined as

$$\Pi_i \stackrel{\text{def}}{=} \{\mathring{\pi}_{i\{f\}} \mid f \text{ is an original primitive site relabeling}\},$$

and the **set of all primitives** is defined, then, as

$$\Pi \ \stackrel{\rm def}{=} \ \bigcup_{i=1}^{n} \Pi_i .$$

For any element $\pi \in \Pi$, the concept of **primitive site relabeling** is defined in the above manner. The concepts of the sets of **initial**, **terminal**, and **all sites** as well as those of the **k-th initial** and **l-th terminal sites** are also extended.

Primitives $\pi_1, \pi_2 \in \Pi$ are called **structurally identical**, denoted $\pi_1 \simeq_{\mathring{\Pi}} \pi_2$, or simply $\pi_1 \simeq \pi_2$, if $\exists \ j \quad \pi_1, \pi_2 \in \Pi_j$ or, equivalently, if there exists a primitive site relabeling $f : \mathrm{Sites}\,(\pi_1) \to SL$ such that $\pi_2 = \pi_1\{f\}$. The corresponding equivalence class $[\pi_1]$ $(= \Pi_j)$ is called a **class**[13] **primitive transformation**, or simply **class primitive** (see Fig. 5). ►

Figure 5: Pictorial illustration of two class primitives (not related to the original primitives in Fig. 4). The circle and the square denote two site types. In other words, letters $\{a, b\}$ and $\{x, y\}$ are names of the variables that are allowed to vary over *non-overlapping* sets of numeric labels (see also footnote 12).

Remark 2. Thus,

$$\forall\, \pi \in \Pi \qquad \exists\, i \ \text{ and site relabeling } \ f \ \text{ such that } \ \pi = \mathring{\pi}_i\{f\}.$$

Moreover, any original primitive $\mathring{\pi}$ can be considered as a *canonical* representative for the corresponding class primitive $[\pi]$.

Remark 3. The index of π_i should not be confused with the index of $\mathring{\pi}_j$.

In concrete examples, instead of $\langle \widehat{\pi}_i, \langle a, b, \ldots, c \rangle, \langle d, e, \ldots, f \rangle \rangle$, the following alternate notation $\widehat{\pi}_i[\, a, b, \ldots, c \,|\, d, e, \ldots, f \,]$ will be used.

Note that site types are permanently associated with sites, while labels are not.

It is useful to note that a primitive name, e.g. $\widehat{\pi}_j$, is supposed to encapsulate the same information as the corresponding class primitive, i.e. $[\pi_1] = \Pi_j$ (see the last paragraph of Def. 4), but in a less explicit form.

[13] The appearance of and meaning of the modifier "class" will become clear later, when the relation between object classes and transformations is established in Def. 18 and in Part III.

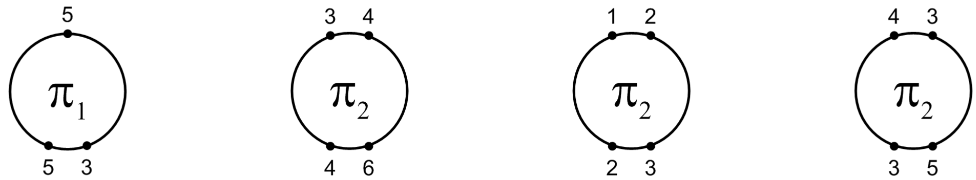


Figure 6: Pictorial illustration of primitives $\widehat{\pi}_1[\,5\,|\,5,3\,]$, $\widehat{\pi}_2[\,3,4\,|\,4,6\,]$, $\widehat{\pi}_2[\,1,2\,|\,2,3\,]$, and $\widehat{\pi}_2[\,4,3\,|\,3,5\,]$. Note that the last three primitives are instances of the same class primitive $[\pi_2]$.

4 Instances of structural history, their composition, and site relabeling

Obviously, the intelligent process must be capable of observing and recording a *consecutive series* of elementary events (depicted in Fig. 7). This series can be thought of as a *macroevent*, or particular instance of "structural history", where a structural history can be thought of as an unlabeled recording/representation of the macroevent.

The following definition[14] should be viewed as a far-reaching structural generalization of the Peano (inductive) construction of natural numbers ([35], [23]).

<u>**Definition**</u> **5.** The set Σ (or, more accurately, $\Sigma_{\mathring{\Pi}}$) of **instances of structural history**, or simply **structs**, is defined inductively[15] as follows. For each $\sigma \in \Sigma$, three sets—$\mathrm{Init}\,(\sigma)$, $\mathrm{Term}\,(\sigma)$, and $\mathrm{Sites}\,(\sigma)$ of **initial sites**, **terminal sites**, and **all sites** of the struct σ—are inductively constructed:

- θ is the **null struct** whose sets of sites are

$$\mathrm{Init}\,(\theta) \;=\; \mathrm{Term}\,(\theta) \;=\; \mathrm{Sites}\,(\theta) \;\stackrel{\mathrm{def}}{=}\; \varnothing$$

- Assuming that $\sigma \in \Sigma$ has been constructed, and given $\pi \in \Pi$ satisfying[16]

$$\mathrm{Sites}\,(\sigma) \cap \mathrm{Sites}\,(\pi) \;=\; \mathrm{Term}\,(\sigma) \cap \mathrm{Init}\,(\pi) \tag{1}$$

  the expression

$$\sigma \dashv \pi$$

  signifies the new struct σ_π, whose sets of sites are constructed as

$$\begin{aligned} \mathrm{Init}\,(\sigma_\pi) &\stackrel{\mathrm{def}}{=} \mathrm{Init}\,(\sigma) \cup [\mathrm{Init}\,(\pi) \setminus \mathrm{Term}\,(\sigma)] & (2)\\ \mathrm{Term}\,(\sigma_\pi) &\stackrel{\mathrm{def}}{=} \mathrm{Term}\,(\pi) \cup [\mathrm{Term}\,(\sigma) \setminus \mathrm{Init}\,(\pi)] & (3)\\ \mathrm{Sites}\,(\sigma_\pi) &\stackrel{\mathrm{def}}{=} \mathrm{Sites}\,(\sigma) \cup \mathrm{Sites}\,(\pi). & (4) \end{aligned}$$

[14] However, see also section 15.1.
[15] See the discussion after this definition.
[16] This condition encapsulates the intuitive pictorial representation of structs and ensures the meaningfulness of (2)–(4), below.

We call σ_π the **continuation of struct σ by primitive π,** where continuation is depicted and could be thought of as an **attachment** of the identical sites in $\mathrm{Term}\,(\sigma)$ and $\mathrm{Init}\,(\pi)$, explicated below. The operation $\dashv$ is called the **continue operation**. We specify struct σ by the following expression encapsulating its construction process

$$\sigma \;=\; [\pi_1 \dashv \pi_2 \dashv \cdots \dashv \pi_t]$$

(see Fig. 7). The order relationship between the indices in the last expression corresponds to the constructive order of the relevant continue operations rather than serving as a *unique* primitive identifier (π_i may coincide with π_j). We assume that this expression is valid for $t = 0$ and, in this case, denotes θ. For the above construction of $\sigma \dashv \pi$, we also use the following terminology: **primitive π is attached to primitive π_i** if, when actually constructing $\sigma \dashv \pi$, *at least one* initial site $\pi(\overline{k})$ of π was attached to one terminal site $\pi_i(\underline{l})$ of π_i. ▶

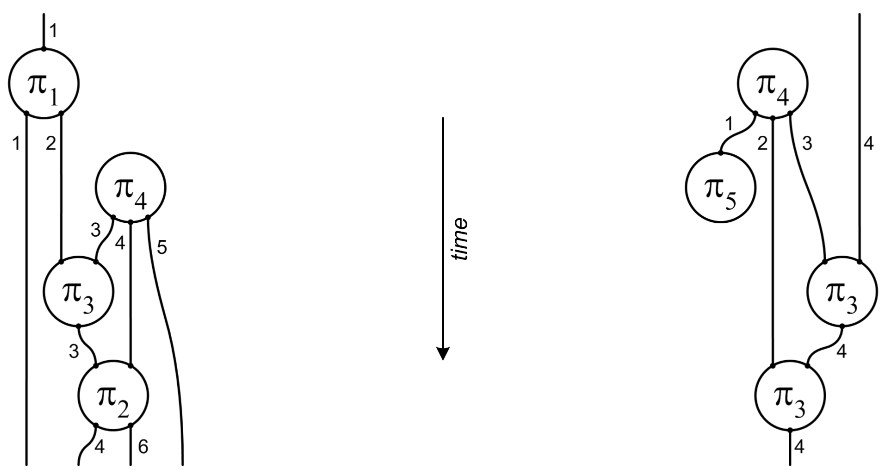


Figure 7: Two structs, where the set of original primitives includes $\mathring{\pi}_1, \mathring{\pi}_2, \mathring{\pi}_3, \mathring{\pi}_4, \mathring{\pi}_5$. The vertical order of primitives corresponds to the constructive (temporal) order of the relevant continue operations.

The proof of the next lemma follows directly from the above definitions.

Lemma 1. Any struct σ, where

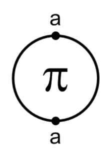
$$\sigma \;=\; [\pi_1 \dashv \pi_2 \dashv \cdots \dashv \pi_t],$$

can also be expressed as

$$\sigma \;=\; [\mathring{\pi}_{i_1}\{f_1\} \dashv \mathring{\pi}_{i_2}\{f_2\} \dashv \cdots \dashv \mathring{\pi}_{i_t}\{f_t\}],$$

where $\mathring{\pi}_{i_j} \in \Pi_{i_j}$ and f_j is an original primitive site relabeling. ■

It is important to note that, *formally*, we view the set Σ not as a derivative of the set of natural numbers, but rather as its precursor. In particular, we view the set of natural numbers as a special case of Σ, where $\mathring{\Pi}$ consists of a single primitive of very simple structure[17]: $\mathring{\pi}[\,a\,|\,a\,]$ (see Fig. 8). Consequently, to achieve a reasonable degree of rigor, we should rely on the appropriate generalization of the Peano axioms (for natural numbers [28]), including the generalization of the induction axiom (see below). In this paper, however, to retain both rigor and accessibility, we adopt instead the following inductive scheme.



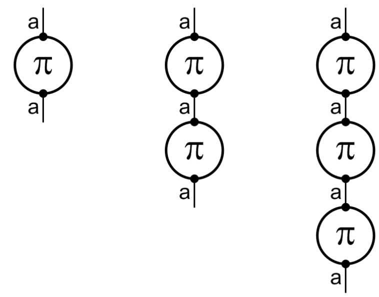


Figure 8: *Left:* the (single) primitive involved in the ETS representation of natural numbers. *Right:* structs representing the natural numbers 1, 2, and 3.

By an inductive definition, or specification, of process $P(\sigma)$[18] that constructs a struct $\sigma \in \Sigma$, we mean the following definition, or specification, scheme:

- construct $P(\theta)$
- assume that $P(\alpha)$ has been constructed and that $\sigma = \alpha \dashv \pi$, then construct $P(\sigma)$.

The above construction relies on the following axiom.

Induction axiom for structs. If Σ' is a subset of Σ satisfying the following conditions

- $\theta \in \Sigma'$
- $\forall\, \sigma,\ \sigma' \in \Sigma$ such that $\sigma' = \sigma \dashv \pi$ for some $\pi \in \Pi$

$$\sigma \in \Sigma' \quad \Rightarrow \quad \sigma' \in \Sigma'$$

then $\Sigma' = \Sigma$. ▶

The next definition introduces two useful kinds of *substructs* of a given struct σ.

<u>Definition</u> 6. For a given struct σ,

$$\sigma \;\; = \;\; [\pi_1 \dashv \pi_2 \dashv \cdots \dashv \pi_t],$$

[17] The set of site labels SL could be chosen to be of cardinality one (abstract natural numbers) or of greater cardinality ("concrete" natural numbers, i.e. counting with sticks, stones, etc.).

[18] See section 2.

and its primitive π_k, **primitive** π_j, **is a successor of** π_k (**in** σ) if there is a subsequence of indices $i_1, i_2, \ldots, i_r$, where $k = i_1 < i_2 < \cdots < i_r = j$, such that

$$\pi_{i_{p+1}} \text{ is attached to } \pi_{i_p} \qquad 1 \le p < r.$$

We also define every primitive to be a successor of itself.

For the above struct σ and a subsequence of its primitives $\langle \pi_{j_1}, \pi_{j_2}, \ldots, \pi_{j_m} \rangle$ such that no primitive is a successor of any other primitive from the set, the minimal struct

$$\sigma(\pi_{j_1}, \pi_{j_2}, \ldots, \pi_{j_m}) \stackrel{\text{def}}{=}$$
$$\left\{ \, [\pi_{k_1} \dashv \cdots \dashv \pi_{k_n}],\ 1 \le k_1 < \cdots < k_n \le t \ \mid\ \forall \pi_{j_i} \text{ all of its successors are in this struct} \right\}$$

is called a **successor substruct of** σ.

If a successor substruct of σ contains the last primitive of σ, π_t, then we call such a substruct a **latest substruct of** σ (see Fig. 9). ▶

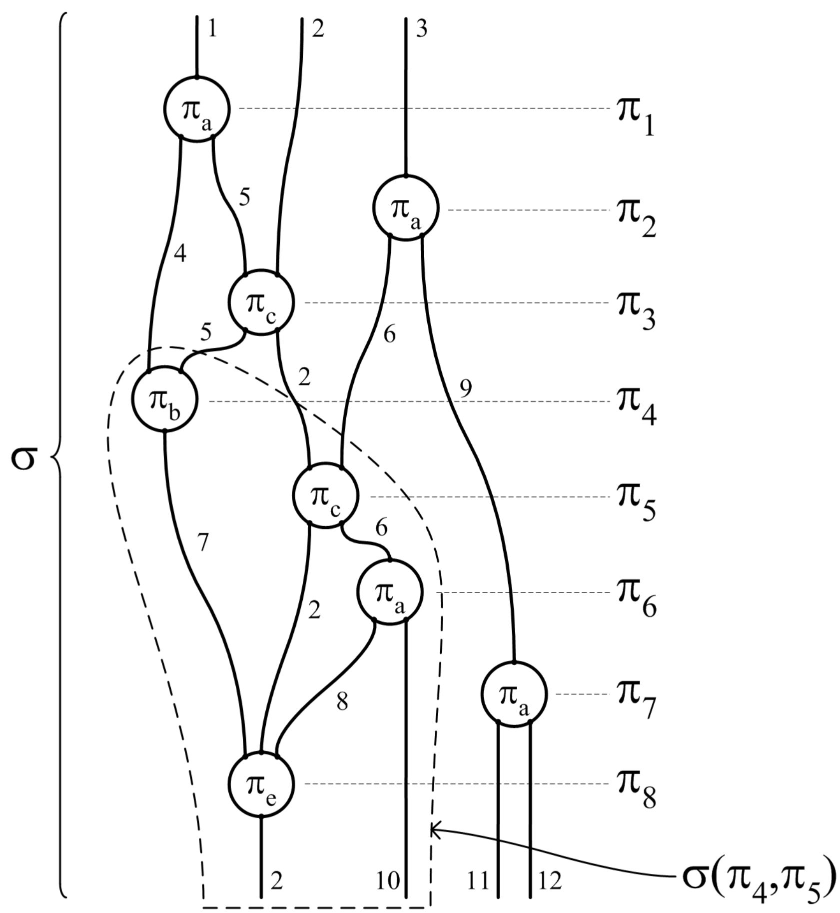


Figure 9: Pictorial illustration of a latest substruct of σ.

Lemma 2. For a given struct σ and two successor substructs $\sigma(\pi_{i_1}, \pi_{i_2}, \ldots, \pi_{i_l})$, $\sigma(\pi_{j_1}, \pi_{j_2}, \ldots, \pi_{j_m})$,

$$\sigma(\pi_{i_1}, \pi_{i_2}, \ldots, \pi_{i_l}) = \sigma(\pi_{j_1}, \pi_{j_2}, \ldots, \pi_{j_m}) \quad \Leftrightarrow \quad \{\pi_{i_1}, \pi_{i_2}, \ldots, \pi_{i_l}\} = \{\pi_{j_1}, \pi_{j_2}, \ldots, \pi_{j_m}\}.$$

■

How do we construct instances of structural history out of existing ones? The following definition introduces the relevant operation of composition of two structs.

Definition 7. Let α and β be structs such that $\mathrm{Init}\,(\beta) \subseteq \mathrm{Term}\,(\alpha)$. If the following inductive construction can be completed[19] then the resulting struct

$$\alpha \triangleleft \beta$$

is called the **composition** of α and β, and we say that β **is composable with** α, denoted $\alpha \prec \beta$. The above inductive construction is specified as follows:

- for $\beta = \theta$,

$$\alpha \triangleleft \theta \overset{\mathrm{def}}{=} \alpha$$

- for $\beta = \gamma \dashv \pi$,

$$\alpha \triangleleft (\gamma \dashv \pi) \overset{\mathrm{def}}{=} (\alpha \triangleleft \gamma) \dashv \pi.$$

►

Note that not every two structs satisfying $\mathrm{Init}\,(\beta) \subseteq \mathrm{Term}\,(\alpha)$ are composable (see Fig. 10).

Lemma 3. The sets of sites for the composition of two structs α and β are

$$\begin{aligned} \mathrm{Init}\,(\alpha \triangleleft \beta) &= \mathrm{Init}\,(\alpha) \cup [\mathrm{Init}\,(\beta) \setminus \mathrm{Term}\,(\alpha)] \\ \mathrm{Term}\,(\alpha \triangleleft \beta) &= [\mathrm{Term}\,(\alpha) \setminus \mathrm{Init}\,(\beta)] \cup \mathrm{Term}\,(\beta) \\ \mathrm{Sites}\,(\alpha \triangleleft \beta) &= \mathrm{Sites}\,(\alpha) \cup \mathrm{Sites}\,(\beta). \end{aligned}$$

■

A pair of uncomposable structs may become composable if we change some of the site labels (recall that site types do not change).

Reading suggestion. We want to emphasize that all definitions related to the various site relabelings should be viewed as *auxiliary* ones from a conceptual point of view and appear for technical reasons. They are introduced here because of *current* formal limitations (and are also related to current hardware/instrumentation technologies that do not allow for direct identification of substructures embedded in some structure[20]). Thus, the above situation has forced our hand into dealing with relabelings, adding unfortunate complexity.

[19] See condition (1) in Def. 5.

[20] We have no doubt that such special-purpose (structural matching) technologies will emerge in the reasonably near future.

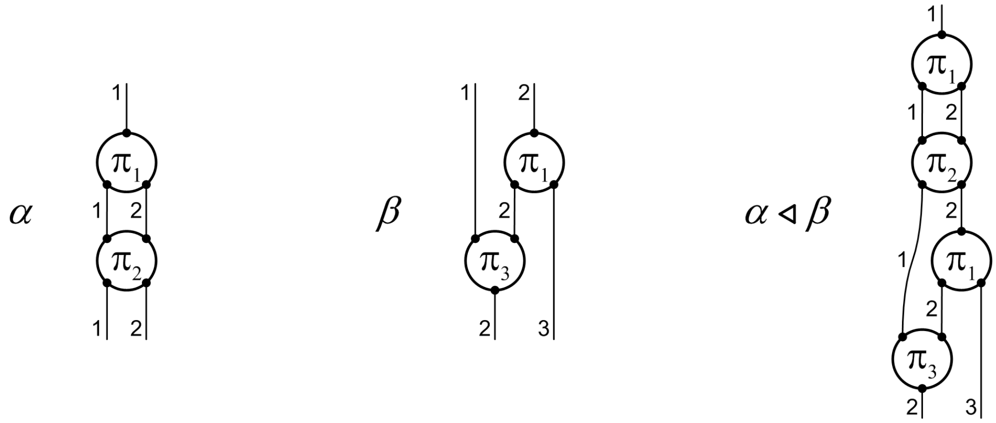


Figure 10: Two structs (α and β) and their composition ($\alpha \triangleleft \beta$). Note that $\beta \triangleleft \alpha$ is not a legal composition.

<u>Definition</u> 8. For a struct $\sigma = [\pi_1 \dashv \pi_2 \dashv \cdots \dashv \pi_t]$, a site relabeling g,

$$g : \text{Sites}\,(\sigma) \to SL,$$

is called a site relabeling of struct σ. Moreover, the struct $\sigma\{g\}$, defined as

$$\sigma\{g\} \stackrel{\text{def}}{=} \left[\pi_1\{g|_{\text{Sites}\,(\pi_1)}\} \dashv \pi_2\{g|_{\text{Sites}\,(\pi_2)}\} \dashv \cdots \dashv \pi_t\{g|_{\text{Sites}\,(\pi_t)}\}\right],$$

is called a **site-relabeled struct** σ (see Fig. 11). ▶

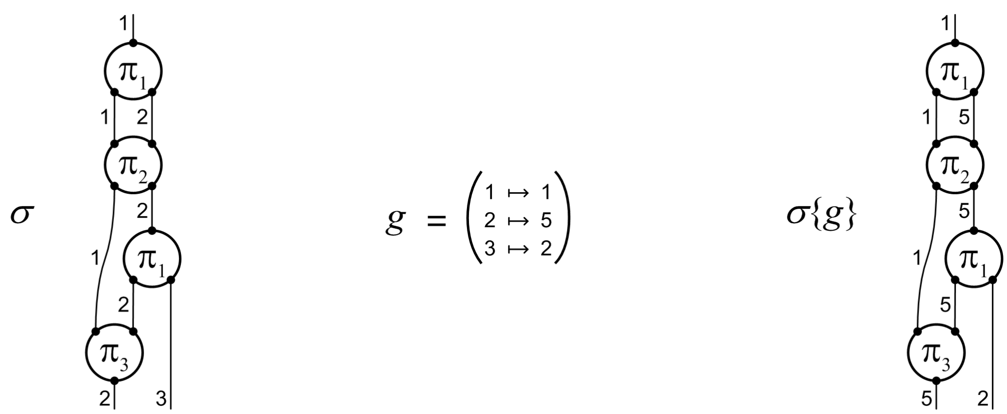


Figure 11: Pictorial illustrations of a struct, a site relabeling mapping, and the corresponding relabeled struct.

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

1. For a struct σ and two site relabelings $f : \text{Sites}\,(\sigma) \to SL$, $g : \text{Sites}\,(\sigma\{f\}) \to SL$, we have

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

2. If $\sigma' = \sigma\{f\}$, then there exists site relabeling $f' : \text{Sites}\,(\sigma') \to SL$ such that $\sigma'\{f'\} = \sigma$.

3. If $\alpha \prec \beta$ and $f : \text{Sites}\,(\alpha \lhd \beta) \to SL$, $f_\alpha = f\big|_{\text{Sites}\,(\alpha)}$, $f_\beta = f\big|_{\text{Sites}\,(\beta)}$ are site relabelings, then $\alpha\{f_\alpha\} \prec \beta\{f_\beta\}$ and
$$\alpha\{f_\alpha\} \lhd \beta\{f_\beta\} \;=\; (\alpha \lhd \beta)\{f\}.$$

5 Extructs

In this section we introduce the concept of "extruct", which may be informally thought of as a particular *fragment* of recent history (which must include the latest event). When the central concept of "transformation" is introduced in the next section, an extruct will specify the "context" of a transformation, i.e. a fragment of a struct which identifies the place where a transformation may originate, while a struct will specify the "body" of a transformation.

We note the tentative nature of the extruct concept: it may change as our understanding of the intelligent process (Part III) evolves.

The next, auxiliary, definition introduces a (partial) encapsulation of a given struct by a graph (see Fig. 12).

Definition 9. The **attachment graph for struct** $\sigma = [\pi_1 \dashv \pi_2 \dashv \cdots \dashv \pi_t]$ is defined as the following directed graph:
$$G_\sigma \;=\; \langle V_\sigma, E_\sigma \rangle,$$
where
$$V_\sigma \;=\; \{v_1, v_2, \cdots, v_t\}$$
$$v_i \text{ corresponds to } \pi_i$$
and
$$\langle v_i, v_j \rangle \in E_\sigma \qquad \text{if, in the inductive construction of } \sigma, \;\; \pi_j \text{ was attached to } \pi_i.$$

►

When no confusions arises, the subscript σ for the attachment graph will be dropped. Note that multiple attachments between two primitives are recorded as a single edge in the attachment graph (see Fig. 12).

The next (also auxiliary) definition introduces several concepts useful for defining one of the main concepts, that of extruct.

Definition 10. An **interfaced struct** is a pair $\langle \sigma, \text{Iface} \rangle$, where σ is a struct
$$\sigma \;=\; [\pi_1 \dashv \pi_2 \dashv \cdots \dashv \pi_t]$$

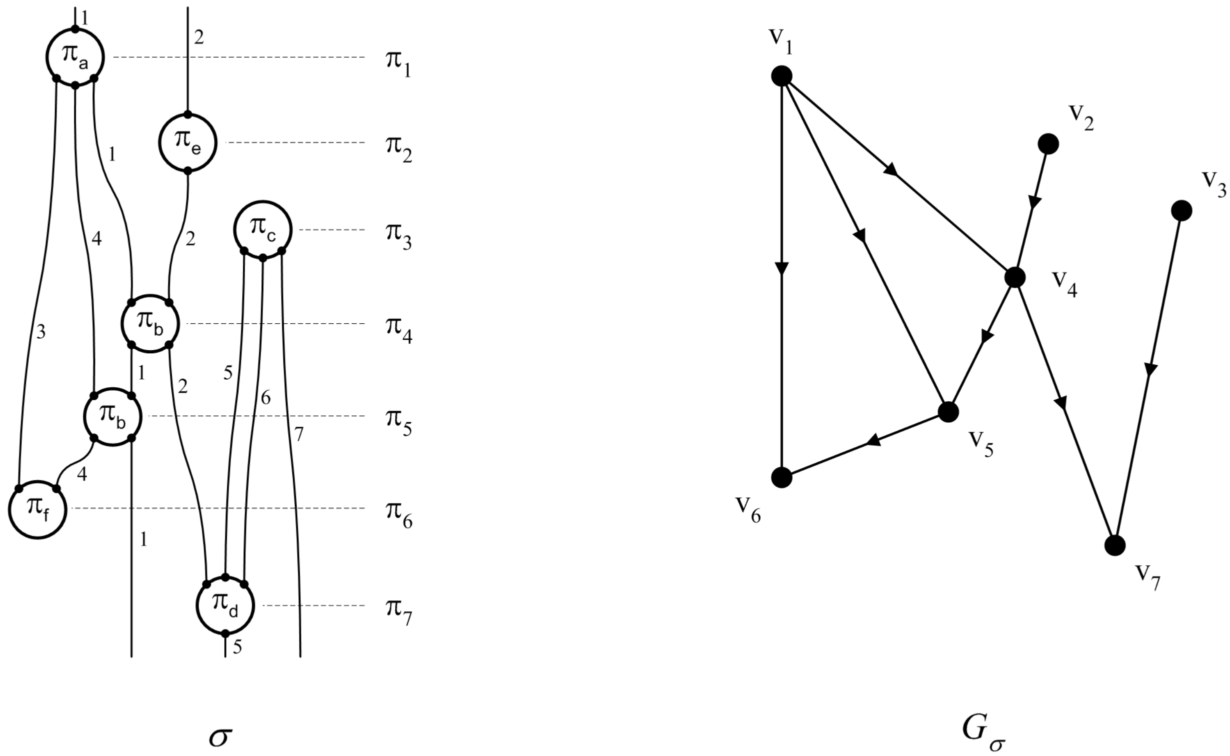


Figure 12: Pictorial illustrations of a struct and the corresponding attachment graph.

and Iface is a subset of $\mathrm{Term}\,(\sigma)$ called the **set of interface sites**. For each primitive π_i in the above σ, $1 \le i \le t$, a **constituent of** $\langle \sigma, \mathrm{Iface}\,\rangle$ is the following 4-tuple

$$\mathfrak{e}^i_\sigma \;\stackrel{\text{def}}{=}\; \langle \pi_i, \mathrm{DIS}_i, \mathrm{DTS}_i, \mathrm{IS}_i \rangle,$$

where

- DIS_i is the set of **detached initial sites**, $\mathrm{DIS}_i \subseteq \mathrm{Init}\,(\pi_i)$, consisting of those initial sites that are not attached to any other primitive;
- DTS_i is the set of **detached terminal sites**, $\mathrm{DTS}_i \subseteq \mathrm{Term}\,(\pi_i) \setminus \mathrm{Iface}\,$, consisting of those terminal sites that are not attached to any other primitive;
- IS_i is the set of **interfaced sites**, $\mathrm{IS}_i \subseteq \mathrm{Term}\,(\pi_i) \cap \mathrm{Iface}\,$, consisting of those terminal sites that are not attached to any other primitive.

▶

For example, for interfaced struct $\big\langle \sigma, \{1,5\} \big\rangle$, where σ is depicted in Fig. 12,

$$\begin{aligned}
\mathfrak{e}_\sigma^1 &= \big\langle\, \pi_1,\ \{1\},\ \varnothing,\ \varnothing \,\big\rangle, \\
\mathfrak{e}_\sigma^2 &= \big\langle\, \pi_2,\ \{2\},\ \varnothing,\ \varnothing \,\big\rangle, \\
\mathfrak{e}_\sigma^3 &= \big\langle\, \pi_3,\ \varnothing,\ \{7\},\ \varnothing \,\big\rangle, \\
\mathfrak{e}_\sigma^4 &= \big\langle\, \pi_4,\ \varnothing,\ \varnothing,\ \varnothing \,\big\rangle, \\
\mathfrak{e}_\sigma^5 &= \big\langle\, \pi_5,\ \varnothing,\ \varnothing,\ \{1\} \,\big\rangle, \\
\mathfrak{e}_\sigma^6 &= \big\langle\, \pi_6,\ \varnothing,\ \varnothing,\ \varnothing \,\big\rangle, \\
\mathfrak{e}_\sigma^7 &= \big\langle\, \pi_7,\ \varnothing,\ \varnothing,\ \{5\} \,\big\rangle.
\end{aligned}$$

The following definition specifies a construction procedure which outputs a σ-extruct, which is, in fact, a particular instance of an extruct (defined immediately after this definition). The procedural nature of this definition, as opposed to the previous ones, should be viewed in light of the "needs" of the intelligent process presented in Part III, which will be actively searching for an occurrence of such an instance in some struct. The procedure consists of the repeated excision of some constituents of a given interfaced struct, together with the corresponding updates of the remaining constituents.

Definition 11. For an interfaced struct $\langle \sigma, \mathrm{Iface} \rangle$,

$$\sigma = [\pi_1 \dashv \pi_2 \dashv \cdots \dashv \pi_t],$$

we perform the following construction procedure. This procedure, called the **σ-related extruct construction procedure**, outputs a **σ-related extruct**, or simply **σ-extruct**. The procedure involves zero or more updates of a tuple (see the next paragraph) $\mathfrak{E}^k$, $k = 0, 1, 2, \ldots, \mathrm{final}$, whose elements are the *current* constituents of $\langle \sigma, \mathrm{Iface} \rangle$ and outputs a triple $\langle \sigma, \mathrm{Iface}, \mathfrak{E} \rangle$, where $\mathfrak{E} = \mathfrak{E}^{\mathrm{final}}$.

First, the procedure constructs the attachment graph G_0 for σ, then initializes the (current) tuple $\mathfrak{E}^l$, i.e. $l = 0$, as follows:

$$\mathfrak{E}^0 \stackrel{\mathrm{def}}{=} \langle \mathfrak{e}_\sigma^1, \mathfrak{e}_\sigma^2, \cdots, \mathfrak{e}_\sigma^t \rangle.$$

In what follows, we use the same notation $\mathfrak{e}_\sigma^i$ to denote the corresponding current (updated) 4-tuple, and the same is true for its components.

Next, $\mathfrak{e}_\sigma^j$ is *allowed*[21] to be excised from the current tuple $\mathfrak{E}^l$ if its current $\mathrm{IS}_j = \varnothing$. This is accomplished by removing $\mathfrak{e}_\sigma^j$ from the current tuple, updating the attachment graph G_l[22], and, for each non-excised $\mathfrak{e}_\sigma^m$,

- if π_m was attached to π_j, updating DIS_m as follows: the new DIS_m is the current DIS_m plus the sites by which π_m was attached to π_j in σ,
- if π_j was attached to π_m, updating DTS_m in a similar manner.

[21] "Allowed" implies a *possibility* that, in practice, depends on the chosen strategy.
[22] The update is accomplished by removing the corresponding vertex and incident edges.

The σ-extruct construction procedure is *allowed* to halt (with the final tuple $\mathfrak{E}$, $\mathfrak{E} = \mathfrak{E}^{\text{final}}$, and the corresponding final attachment graph G, $G = G_{\text{final}}$), if each connected component G' of G has the following property:

$$\bigcup_{v_i \in V_{G'}} \mathrm{IS}_i \neq \varnothing.$$

Thus, the above σ-extruct is the resulting triple $\langle \sigma, \mathrm{Iface}, \mathfrak{E} \rangle$, denoted ε_σ,

$$\varepsilon_\sigma \stackrel{\text{def}}{=} \langle \sigma, \mathrm{Iface}, \mathfrak{E} \rangle,$$

where set Iface is called the **set of interface sites of** ε_σ. Obviously, if $\mathrm{Iface} = \varnothing$, then the resulting σ-extruct

$$\varepsilon_\sigma = \langle \sigma, \varnothing, \varnothing \rangle$$

is called the **null σ-extruct**. The set of all σ-extructs for an interfaced struct $\langle \sigma, \mathrm{Iface} \rangle$ is denoted $\mathcal{E}_{\langle \sigma, \mathrm{Iface} \rangle}$. The set of all σ-extructs for all interfaced structs is denoted $\mathcal{E}$. ▶

Figure 13 illustrates three *possible* outputs of the above construction procedure for σ from Fig. 12.

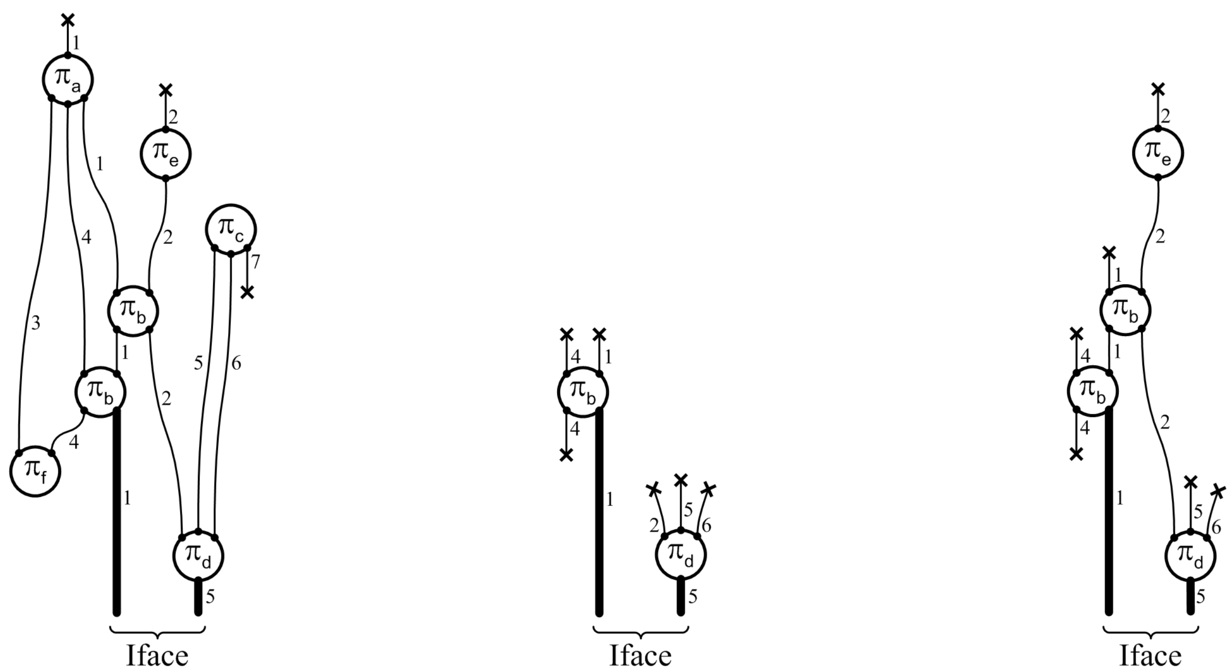


Figure 13: Pictorial illustration of some σ-extructs for the interfaced struct $\langle \sigma, \mathrm{Iface} \rangle$, where σ is shown in Fig. 12. (Heavy lines identify interface sites, crosses identify detached initial/terminal sites.)

Remark 4. Note that the above σ-extruct construction procedure may be specified in greater detail, but such a specification is *application-dependent*.[23]

[23] See also Remark 5 in Section 6.

The central concept of this section, introduced next, is a generalization of the σ-related extruct obtained by removing the dependence on struct σ, which is one of the input parameters of the corresponding construction procedure in Def. 11.

<u>**Definition**</u> **12.** An **extruct** ε is defined as an equivalence class for the following equivalence relation $\sim$ on the set $\mathcal{E}$. Let

$$\varepsilon_{\sigma_1} \; = \; \langle \sigma_1, \text{Iface}_1, \mathfrak{E}_1 \rangle \qquad \text{and} \qquad \varepsilon_{\sigma_2} \; = \; \langle \sigma_2, \text{Iface}_2, \mathfrak{E}_2 \rangle,$$

then

$$\varepsilon_{\sigma_1} \sim \varepsilon_{\sigma_2} \quad \Longleftrightarrow \quad \text{Iface}_1 \, = \, \text{Iface}_2 \quad \text{and} \quad \mathfrak{E}_1 \, = \, \mathfrak{E}_2.$$

We also use the following notation

$$\varepsilon \;\stackrel{\text{def}}{=}\; \langle \text{Iface}, \mathfrak{E} \rangle,$$

where

$$\text{Iface} \;\stackrel{\text{def}}{=}\; \text{Iface}_1,$$

$$\mathfrak{E} \;\stackrel{\text{def}}{=}\; \mathfrak{E}_1.$$

In this context, the set Iface is called the **set of interface sites of extruct** ε, and the set

$$\text{Sites}(\varepsilon) \;\stackrel{\text{def}}{=}\; \bigcup \text{Sites}(\pi_i),$$

where each π_i corresponds to $\mathfrak{e}_\sigma^i$ in tuple $\mathfrak{E}$, is called the **set of all sites of extruct** ε. ▶

When pictorially illustrating an extruct, we simply use the pictorial illustration of a corresponding σ-extruct.

We now introduce the concept of relabeling for an extruct.

<u>**Definition**</u> **13.** For an extruct ε, a site relabeling g,

$$g : \text{Sites}(\varepsilon) \rightarrow SL,$$

is called a **site relabeling of extruct** ε. Moreover, the extruct

$$\varepsilon\{g\} \;\stackrel{\text{def}}{=}\; \langle g(\text{Iface}), \mathfrak{E}\{g\} \rangle,$$

where $\mathfrak{E}\{g\}$ is obtained from $\mathfrak{E}$ by replacing $\mathfrak{e}_\sigma^i = \langle \pi_i, \text{DIS}_i, \text{DTS}_i, \text{IS}_i \rangle$ (for an appropriate σ) with

$$\mathfrak{e}_\sigma^i\{g\} \quad = \quad \langle \pi_i\{g|_{\text{Sites}(\pi_i)}\}, g(\text{DIS}_i), g(\text{DTS}_i), g(\text{IS}_i) \rangle,$$

is called a **site-relabeled extruct** ε. ▶

6 Transformations and supertransformations

This section might be viewed as the culmination of Part II: in it, we introduce almost all central concepts of the proposed framework.

The following definition introduces the first of the three *most* fundamental concepts of the ETS framework, that of “transformation”, which can be thought of as a representational module. A transformation embodies a formative dependence between its two constituent submodules: an extruct and the struct that is attached to it (see Fig. 14).

Definition 14. A **transformation**, or simply **transform**, is a pair

$$\tau \quad = \quad \langle \varepsilon, \beta \rangle$$

where extruct $\varepsilon = \langle \mathrm{Iface} \, , \mathfrak{E} \rangle$ and struct β satisfy

$$\mathrm{Iface} \quad = \quad \mathrm{Init}\,(\beta) \quad = \quad \mathrm{Sites}\,(\varepsilon) \cap \mathrm{Sites}\,(\beta).$$

We call ε the **context of transform** τ, denoted $\mathrm{cntx}(\tau)$, and β the **body of transform** τ, denoted $\mathrm{body}(\tau)$. The **set of all sites of transform** τ is defined as

$$\mathrm{Sites}\,(\tau) \ \stackrel{\text{def}}{=} \ \mathrm{Sites}\,(\varepsilon) \cup \mathrm{Sites}\,(\beta).$$

The set of all transforms is denoted T. ▶

Remark 5. The current asymmetry between the concepts of context and body is related to the fact that the context of a transform has to be “detected” within a given struct, while the transform’s body is “grown” from the identified context. Context detection is realized via the σ-related extruct construction procedure (Def. 11). The nature of the process involved in the growth of the body is sketched in Part III (however, see section 15).

From an applied point of view, it is useful to think of a struct as being “formed”, roughly speaking, by a series of transformations, since, in reality, every struct stands for an “object” from some *class* of objects (see also Defs. 18, 29).

As always, we now need to be able to relabel a transform’s sites.

Definition 15. For a transform $\tau = \langle \varepsilon, \beta \rangle$, a site relabeling h,

$$h : \mathrm{Sites}\,(\tau) \rightarrow SL,$$

is called a **site relabeling of transform** τ. Moreover, the transform

$$\tau\{h\} \ \stackrel{\text{def}}{=} \ \big\langle \varepsilon\{h|_{\mathrm{Sites}\,(\varepsilon)}\}, \beta\{h|_{\mathrm{Sites}\,(\beta)}\} \big\rangle,$$

is called a **site-relabeled transform** τ. ▶

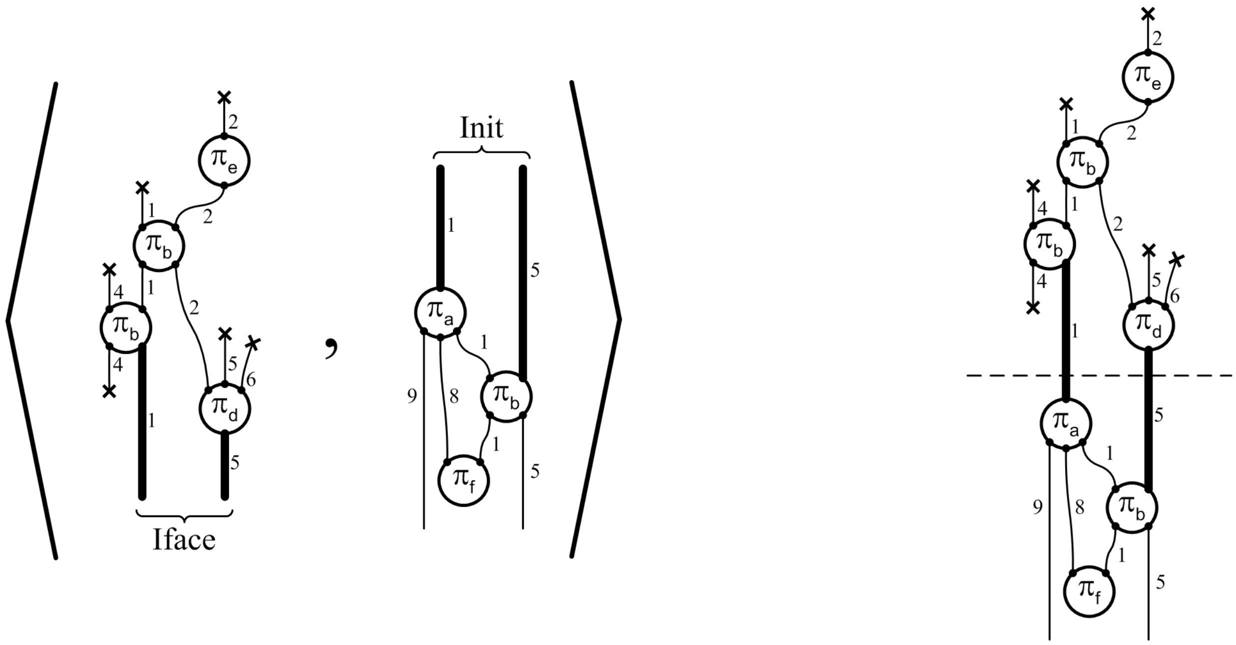


Figure 14: An example of a transform whose context is an extruct induced by the last σ-extruct in Fig. 13. The right hand side depicts the "assembled" transform corresponding to a more appropriate interpretation/understanding of the transform.

We are now ready to introduce a generalization of the transformation concept, that of supertransformation, which is achieved by uniting several related transformations (see Fig. 15). Although at this moment the role of supertransform is not clarified, we will do so in the next section, at which point its meaning will also become clearer. Unfortunately, we must admit that the following definition of this concept—but not the concept itself—is a transient one (see also the Important remark on p. 36).

<u>**Definition**</u> **16.** A **supertransformation**, or simply **supertransform**, is a pair

$$\boldsymbol{\tau} \stackrel{\text{def}}{=} \langle E, B \rangle,$$

where

$$E = \{\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_p\} \qquad \varepsilon_i = \langle \text{Iface}_{\,i}, \mathfrak{E}_i \rangle,$$
$$B = \{\beta_1, \beta_2, \cdots, \beta_q\},$$

if the following conditions hold

$$\forall\, i, j, k \qquad \begin{aligned} \text{Init}\,(\beta_i) = \text{Init}\,(\beta_j) = \text{Iface}_{\,k} = \text{Sites}\,(\varepsilon_k) \cap \text{Sites}\,(\beta_i) \\ \text{Term}\,(\beta_i) = \text{Term}\,(\beta_j)\,. \end{aligned}$$

The **constituent transform set** for a supertransform $\boldsymbol{\tau}$ is defined as the set of all transforms specified by the elements of the Cartesian product $E \times B$.

It is convenient for us to *blur the distinction* between the pair $\langle E, B\rangle$ and the product $E \times B$, and to refer to both of them as the supertransform $\boldsymbol{\tau}$. Thus the following notation will be used: $\tau = \langle \varepsilon, \beta\rangle, \quad \tau \in \boldsymbol{\tau}$.

The **set of all sites of supertransform $\boldsymbol{\tau}$** is defined as

$$\text{Sites}\,(\boldsymbol{\tau}) \stackrel{\text{def}}{=} \bigcup_{\tau \in \boldsymbol{\tau}} \text{Sites}\,(\tau).$$

The set of all supertransforms is denoted $\boldsymbol{T}$. ▶

It is easy to see that the concept of a "constituent transform" is meaningful, i.e. the conditions of Def. 14 are satisfied. Moreover, in the terminology of Def. 16, all constituent transforms share interface sites.

A useful visualization aid for a supertransform $\boldsymbol{\tau}$ is the rectilinear table of its constituent transforms, as shown in Fig. 15.

It is useful to note that there are supertransforms with null contexts: all of its contexts are null, and all of its bodies have no initial sites, i.e. the table in Fig. 15 collapses to a single row of bodies.

Again, we need to be able to relabel supertransforms.

Definition 17. For a supertransform $\boldsymbol{\tau} = \langle E, B\rangle \quad (= E \times B)$, a site relabeling h

$$h : \text{Sites}\,(\boldsymbol{\tau}) \to SL,$$

is called a **site relabeling of supertransform $\boldsymbol{\tau}$**. Moreover, the supertransform

$$\boldsymbol{\tau}\{h\} \stackrel{\text{def}}{=} \left\{ \tau\{h|_{\text{Sites}\,(\tau)}\} \,\middle|\, \tau \in \boldsymbol{\tau} \right\}$$

is called a **site-relabeled supertransform $\boldsymbol{\tau}$**. ▶

The following definition is the second of the three *most* fundamental concepts of the ETS framework; it is a simple/natural generalization of the supertransform concept (see Fig. 16).

Definition 18. A **class supertransform** is defined as an equivalence class for the following equivalence relation $\approx$ on the set $\boldsymbol{T}$ of all supertransforms. Let $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2$ be supertransforms, then

$$\boldsymbol{\tau}_1 \approx \boldsymbol{\tau}_2 \quad \Longleftrightarrow \quad \boldsymbol{\tau}_2 = \boldsymbol{\tau}_1\{g\}$$

for some supertransform site relabeling mapping g. The notation $[\boldsymbol{\tau}]$ is used to denote a class supertransform containing $\boldsymbol{\tau}$ and is called the class *induced* by $\boldsymbol{\tau}$. ▶

It is useful to think of a class supertransform as a "structural" *description* of a class of objects that allows for context- and body-related noise variations, including some transformations that only account for noise (see also Def. 29 where further generalization is introduced). We also need a particular "canonical" supertransform *standing for* the class supertransform.

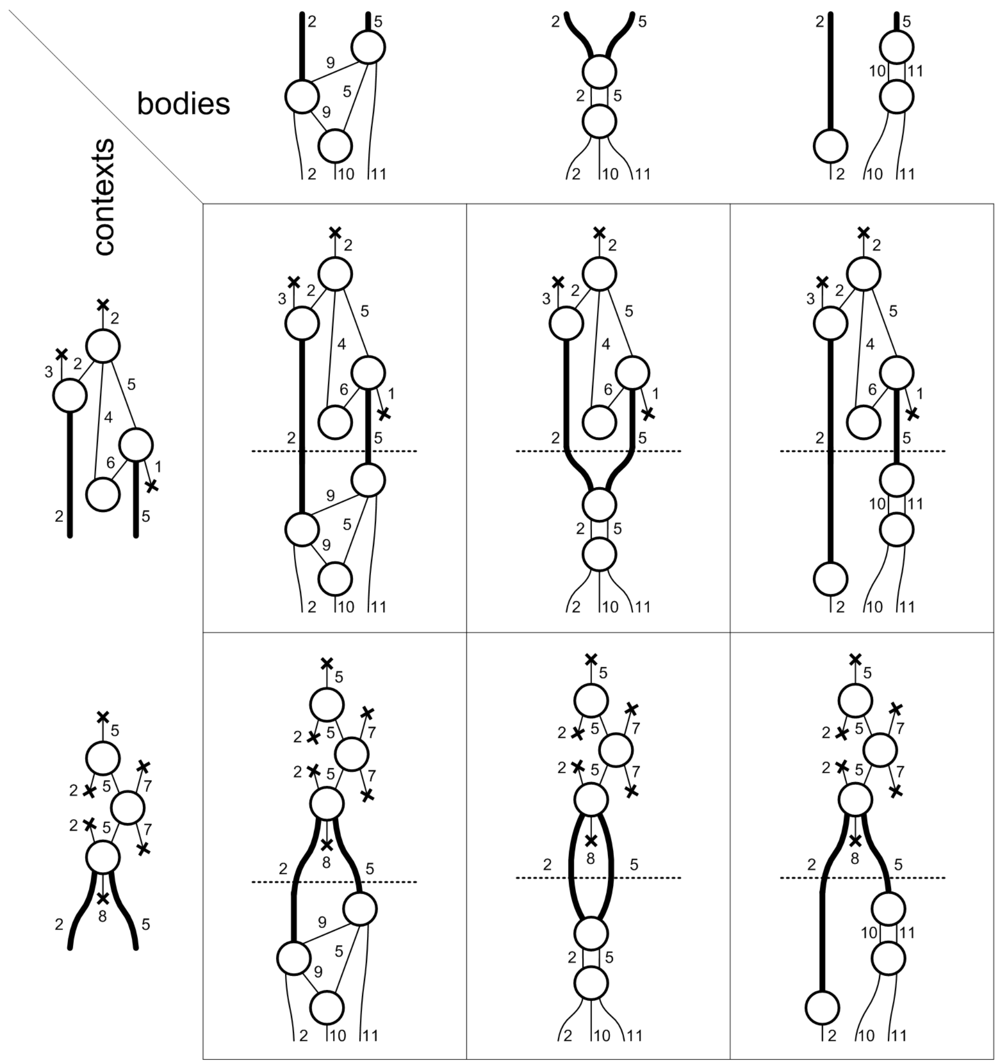


Figure 15: Pictorial illustration of a supertransform. Note that all contexts have the same interface sites and all bodies have the same initial and terminal sites.

Lemma 4. For any class supertransform $[\boldsymbol{\tau}]$ there exists an efficient, implementation-dependent algorithm that uniquely constructs a particular supertransform $\mathring{\boldsymbol{\tau}}$, $\mathring{\boldsymbol{\tau}} \in [\boldsymbol{\tau}]$. ■

The lemma leads naturally to the next definition (see also Remark 2, p. 16).

<u>Definition</u> 19. The unique supertransform $\mathring{\boldsymbol{\tau}}$ constructed in the above lemma is called the **canonical supertransform** for class supertransform $[\boldsymbol{\tau}]$. ►

The existence of canonical supertransforms simplifies the algorithmic processing and storage of class supertransforms and also facilitates a transition between two adjacent levels of representation (see the next section). Further, any supertransform $\boldsymbol{\upsilon} \in [\boldsymbol{\tau}]$ can now be specified simply by providing a site relabeling h such that $\boldsymbol{\upsilon} = \mathring{\boldsymbol{\tau}}_{\{h\}}$.

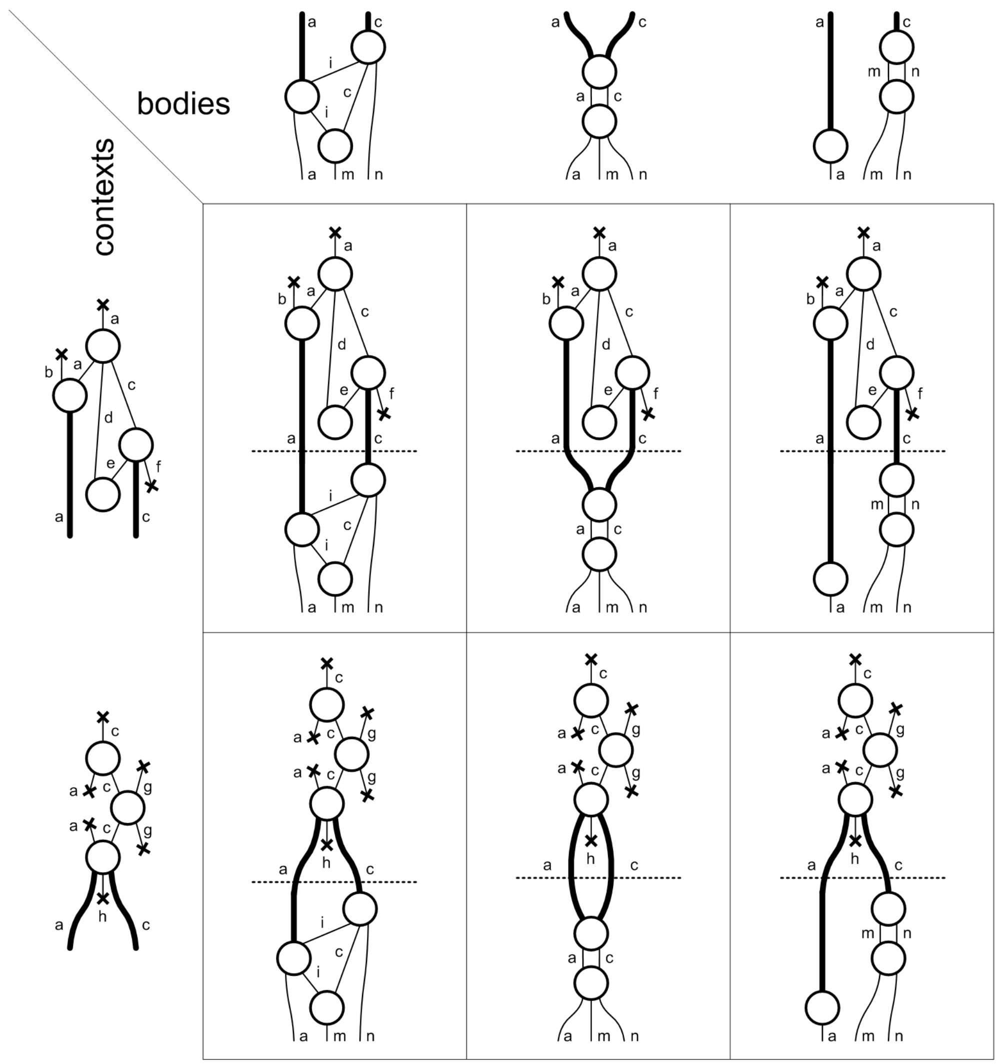


Figure 16: Pictorial illustration of a class supertransform induced by the supertransform depicted in Fig. 15. Each letter is the name of a variable that is allowed to vary over numeric labels of the same type, as explained in the caption of Fig. 5 (site types are not shown).

How does one expand a class supertransform on the basis of a given transform υ whose body is present in that class supertransform? For the set of bodies in the canonical supertransform (from the class supertransform) that are relabelings of the body of υ, we add the same number of contexts, each of which is an appropriately relabeled context of υ.

<u>Definition</u> 20. For a given transform υ,

$$\upsilon \;=\; \langle \zeta, \gamma \rangle,$$

and class supertransform $[\boldsymbol{\tau}]$, with canonical supertransform $\mathring{\boldsymbol{\tau}} = \langle E, B \rangle$, let set $\mathcal{G}$ be the set of all the relabelings g of body γ such that $\gamma_{\{g\}}$ is in B:

$$\mathcal{G} \;\stackrel{\text{def}}{=}\; \big\{ g : \text{Sites}\,(\gamma) \to SL \,|\, \gamma_{\{g\}} \in B \big\}.$$

Moreover, let set $\mathcal{H}$ be a set of relabelings of υ that are extensions of g's and such that for each $g \in \mathcal{G}$ there exists unique $h \in \mathcal{H}$ satisfying

$$h|_{\mathrm{Sites}\,(\gamma)} = g$$

$$\mathrm{Sites}\,(\mathring{\boldsymbol{\tau}}) \cap h\big(\mathrm{Sites}\,(\zeta) \setminus \mathrm{Sites}\,(\gamma)\big) = \varnothing.$$

If $\mathcal{G} \neq \varnothing$, let

$$E_{\mathrm{exp}} \ \stackrel{\mathrm{def}}{=}\ E \cup \big\{\zeta_{\{h|_{\mathrm{Sites}\,(\zeta)}\}} \mid h \in \mathcal{H}\big\}.$$

The **context expansion of class supertransform $[\boldsymbol{\tau}]$ w.r.t. transform** υ, denoted $[\boldsymbol{\tau}, \mathrm{cntx}(\upsilon)]$, is defined as the class supertransform induced by the supertransform

$$\boldsymbol{\tau}_{\mathrm{cntx}(\upsilon)} \ \stackrel{\mathrm{def}}{=}\ \langle E_{\mathrm{exp}}, B\rangle.$$

The concept of the **body expansion of class supertransform $[\boldsymbol{\tau}]$ w.r.t. transform** υ, denoted $[\boldsymbol{\tau}, \mathrm{body}(\upsilon)]$, is defined similarly by exchanging the roles of ζ and γ in the above. ▶

7 Transition to a new level of representation

What necessitates the transition (within an information processing system) to a new level of representation is the need to deal more effectively with the complexity of event representation. In AI this was often called "chunking" [37]. Within the ETS model, such a transition consists of the construction of a new (next-level) set of primitives, which can then be used constructively in the usual manner (including construction of structs, extructs, transforms, etc.).

How does one encapsulate the information contained in a class supertransform in the form of a new primitive in order to legitimately reduce the complexity of an event representation? As a transitory working version, we propose the following postulate to deal with this question. This postulate is based on a considerable oversimplification related to the fact that we have chosen to almost completely ignore the structure of the contexts and the bodies of the supertransform[24].

Level ascension postulate. The class of (context-sensitive) macroevents corresponding to a canonical supertransform may be adequately represented at the next level by a new (original) primitive obtained by completely shrinking that supertransform's contexts and by dropping the internal structure of the supertransform's bodies in the manner described in Def. 21. ▶

Notational convention 1. From this section onwards, next-level notations will be denoted with the addition of a superscript prime to the corresponding present-level notation.

[24] However, see section 15.2.

The following definition is a direct consequence of the above postulate and Def. 1 (including the notation in the definition).

Definition 21. Assume that we have fixed a set **TS** of class supertransforms,

$$\mathbf{TS} \;=\; \{[\boldsymbol{\tau}_1], [\boldsymbol{\tau}_2], \cdots, [\boldsymbol{\tau}_m]\},$$

called a **transformation system**. Define three sets[25]

$$\begin{array}{rcll} \widehat{\Pi}' & \stackrel{\text{def}}{=} & \left\{\widehat{[\boldsymbol{\tau}_1]}, \widehat{[\boldsymbol{\tau}_2]}, \cdots, \widehat{[\boldsymbol{\tau}_m]}\right\} & \text{of } \textbf{next-level primitive names}, \\ SL' & \stackrel{\text{def}}{=} & SL & \text{of } \textbf{next-level site labels}, \\ ST' & \stackrel{\text{def}}{=} & ST & \text{of } \textbf{next-level site types}. \end{array}$$

We now introduce a set of **next-level original primitives** $\mathring{\Pi}'$ for which each of its elements $\mathring{\pi}'_i$ is constructed as follows:

$$\mathring{\pi}'_i \;\stackrel{\text{def}}{=}\; \left\langle \widehat{[\boldsymbol{\tau}_i]}, \mathrm{INIT}_i, \mathrm{TERM}_i \right\rangle$$

where, for $\mathring{\boldsymbol{\tau}}_i = \langle E_i, B_i \rangle$ with $B_i = \{\beta_{i1}, \beta_{i2}, \ldots, \beta_{iq_i}\}$,

$$]\mathrm{INIT}_i[\;\stackrel{\text{def}}{=}\; \mathrm{Init}\left(\beta_{i1}\right),$$

$$]\mathrm{TERM}_i[\;\stackrel{\text{def}}{=}\; \mathrm{Term}\left(\beta_{i1}\right),$$

and the corresponding linear orders are induced in an appropriate manner[26] (see Fig. 17). In addition, we define a **next-level site type mapping** $\mathrm{TYPE}' : SL' \to ST'$ to be the same as mapping TYPE in Def. 1. ▶

It is important to note that, under the adopted scheme for level ascension, the site types for the new primitives come from the same set ST. In light of the above-mentioned oversimplification, this situation is not satisfactory: it would be desirable not to refer to the previous set ST, but to introduce a new one. However, another way to accomplish this is to develop an approach for the introduction of new (next-level) sites based on some existing structural features at the present level, e.g. next-level sites as current-level primitives[27].

Definition 22. We define the next-level analogues of the concepts introduced in Def. 2 (initial sites, terminal sites, all sites, etc.) in exactly the same manner. ▶

---

[25] $\widehat{[\boldsymbol{\tau}_i]}$, $1 \le i \le m$, could be thought of as denoting the *"name" given to the class supertransform* $[\boldsymbol{\tau}_i]$, which is *inherited by* $\mathring{\boldsymbol{\tau}}_i$.

[26] This order is induced based on both the constructive order of the primitives in the first body of $\mathring{\boldsymbol{\tau}}_i$ as well as on the orders of the sites in each of those primitives.

[27] For a substantially different treatment of this issue, see [21].

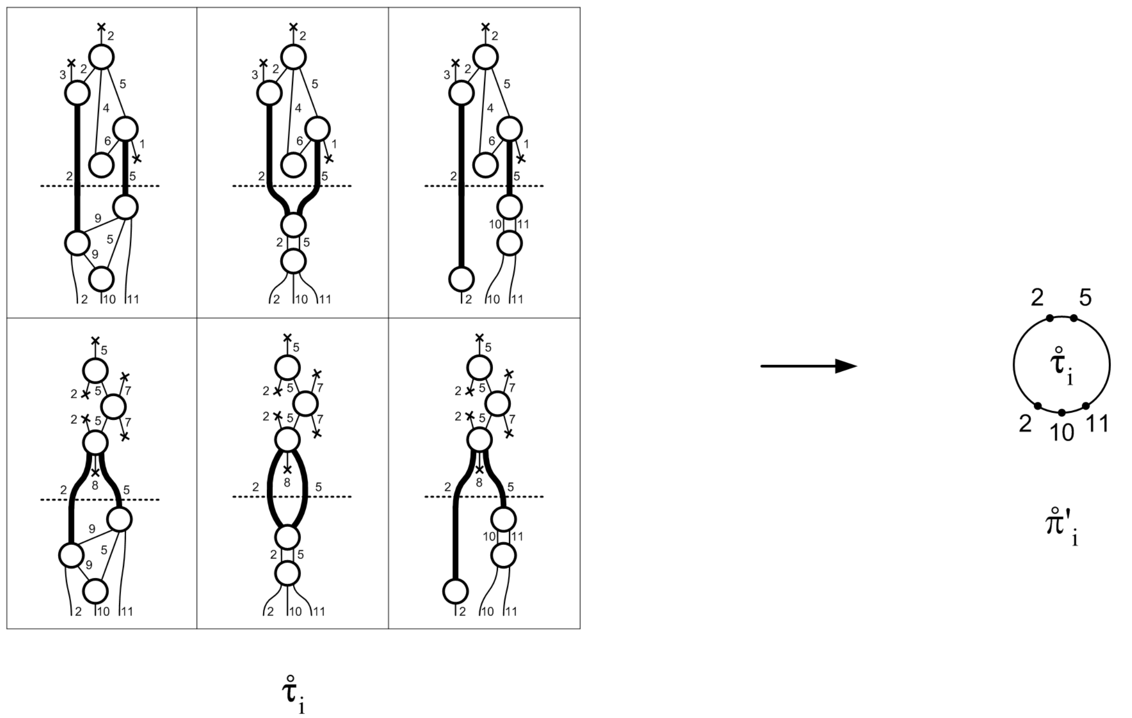


Figure 17: A canonical supertransform (from a class supertransform taken from a transformation system) and the corresponding next-level original primitive. Note that, as previously mentioned in the caption of Fig. 4, the symbol $\widehat{\ }$ in the depiction of the (next-level) original primitive is dropped.

Definition 23. We define the next-level analogues of the concepts introduced in Def. 4 (original primitive site relabeling, primitives, etc.) in exactly the same manner. ▶

We are now ready to state an improved version of the above correspondence postulate.

Refined level ascension postulate. The ascension to the next level is based on the following basic correspondence[28]

$$\mathring{\boldsymbol{\tau}}_i \;\to\; \mathring{\pi}'_i \tag{5}$$

$$\forall\, \boldsymbol{\tau},\ \boldsymbol{\tau} = \mathring{\boldsymbol{\tau}}_i\{h\} \qquad \boldsymbol{\tau} \;\to\; \pi' = \mathring{\pi}'_i\{h|_{\mathrm{Sites}\,(\mathring{\pi}'_i)}\} \tag{6}$$

where the notation in (5) comes from Defs. 19, 21 and the notation in (6) comes from Defs. 19, 23[29].

All other correspondences between adjacent levels are implications of the above correspondence. For example, a next-level struct corresponds to a previous-level superstruct composed of structs (bodies), where each body is that of one of the transforms from the corresponding supertransform. ▶

[28] The arrow $\to$ denotes ascension to the next level. See also Remark 2 in Sec. 3.

[29] See also the paragraph following Def. 19.

Important remark. In view of the above correspondence (5), we note that the restrictions imposed on a supertransform in Def. 16 are not quite sufficient: it is intuitively clear that the supertransform must satisfy some additional constraints ensuring the closer interrelationship of its constituent transform's bodies (see Fig. 18).

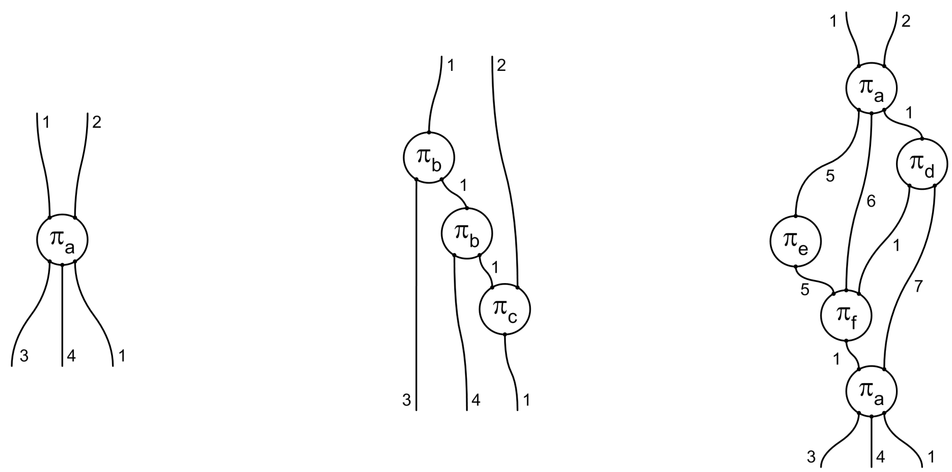


Figure 18: Three "consistent", but otherwise quite unrelated bodies.

Finally, we can encapsulate the entire developed mathematical structure as a single entity in the following definition.

<u>Definition</u> **24.** A **(single-level) inductive structure** is a pair

$$\langle \mathring{\Pi}, \mathbf{TS} \rangle,$$

where $\mathring{\Pi}$ is a set of original primitives and $\mathbf{TS}$ is a transformation system. However, as was mentioned above, the latter pair also signifies all relevant concepts, such as structs, extructs, etc.

A **multi-level inductive structure** (with l levels) $\mathbb{MIS}$ is an l-tuple

$$\mathbb{MIS} \overset{\text{def}}{=} \left\langle \langle \mathring{\Pi}, \mathbf{TS} \rangle, \langle \mathring{\Pi}', \mathbf{TS}' \rangle, \cdots, \langle \mathring{\Pi}^{(l-1)}, \mathbf{TS}^{(l-1)} \rangle \right\rangle,$$

where $\mathbf{TS}^{(l-1)} = \varnothing$, $\mathbf{TS}^{(k)}$ is the transformation system for the set of original primitives $\mathring{\Pi}^{(k)}$, and every consecutive pair of inductive structures satisfies the (refined) level ascension postulate (see Figs. 19, 20). For the k-th level inductive structure $\left\langle \mathring{\Pi}^{(k)}, \mathbf{TS}^{(k)} \right\rangle$ in $\mathbb{MIS}$, we use the notation

$$\mathbb{MIS}(k) \overset{\text{def}}{=} \langle \mathring{\Pi}^{(k)}, \mathbf{TS}^{(k)} \rangle \qquad k = 0, 1 \cdots, l-1$$

$$\boldsymbol{\tau}^{(k)} \to \pi^{(k+1)} \qquad k = 0, 1 \cdots, l-2.$$

►

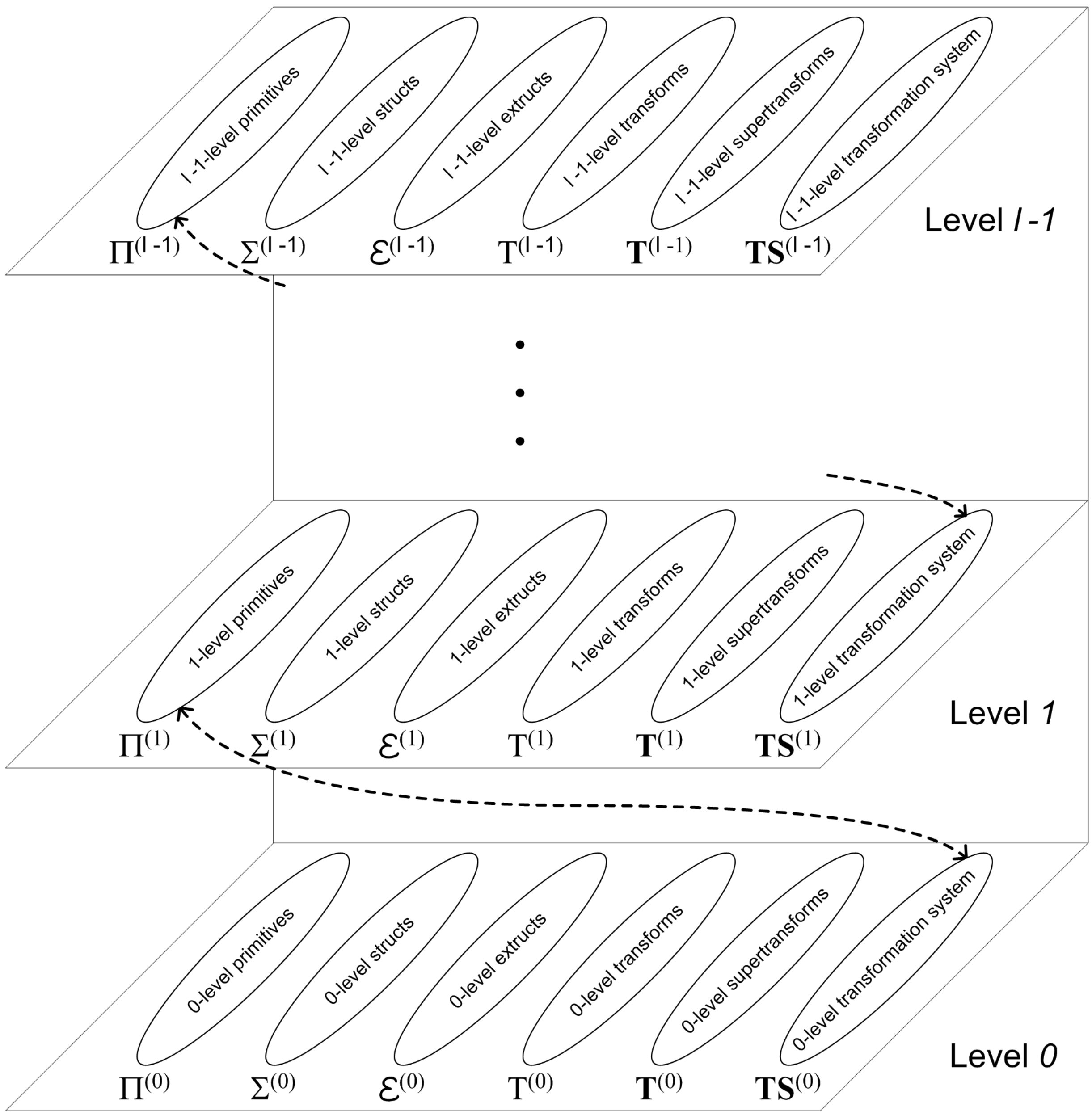


Figure 19: Schematic representation of a multi-level inductive structure with l levels.

Notational convention 2. For a transform τ from $\boldsymbol{\tau}$, where $\boldsymbol{\tau} \in [\boldsymbol{\tau}] \in \mathbf{TS}$, we use the following simplified notation $\tau \in \boldsymbol{\tau} \in \mathbf{TS}$. Similarly, for the latter supertransform $\boldsymbol{\tau}$, we write $\boldsymbol{\tau} \in \mathbf{TS}$.

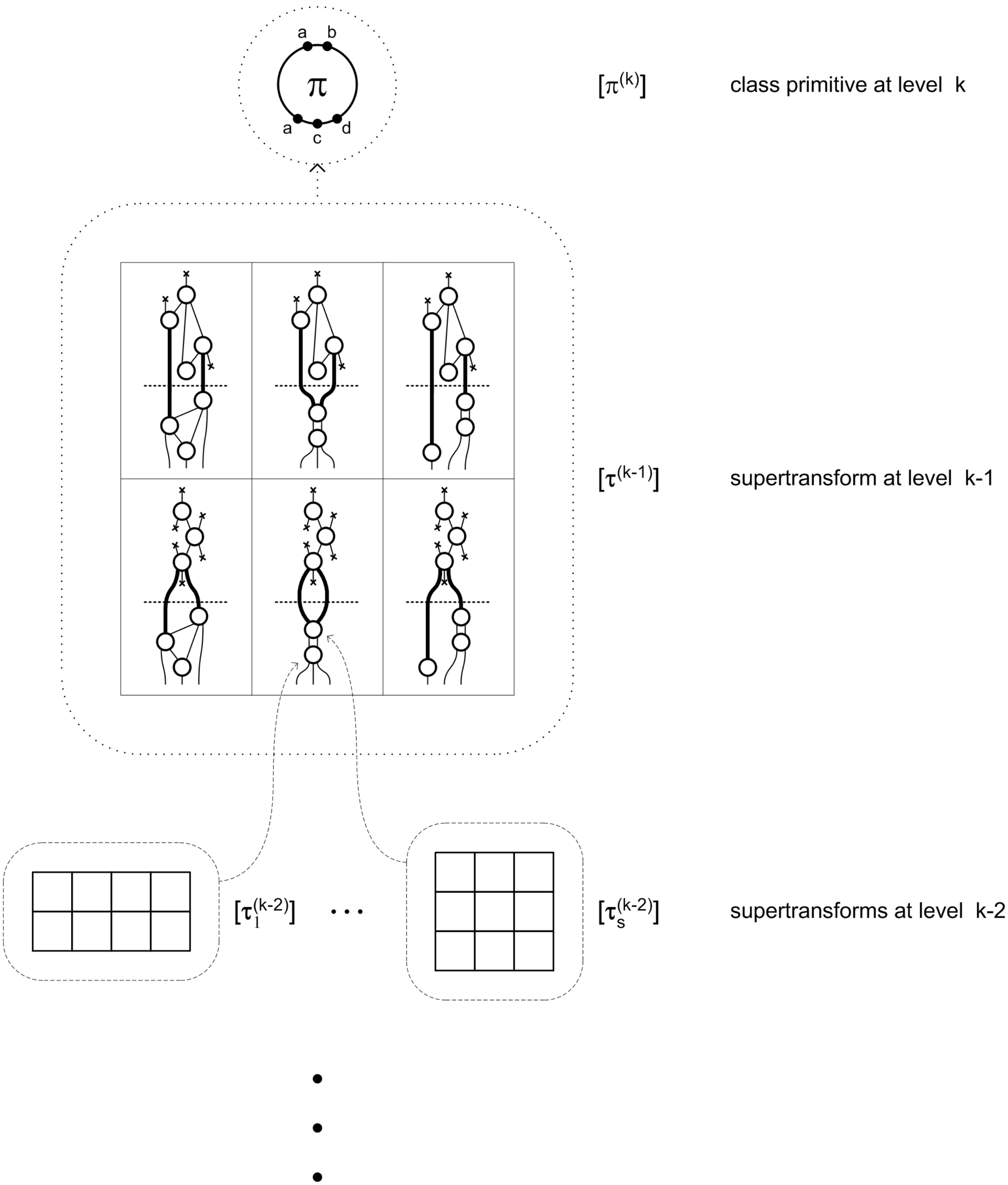


Figure 20: Pyramid view (partial) of a k-th level class supertransform: the pyramid should be thought of as being formed by the subordinate class supertransforms.

Part III

The intelligent process:

a provisional sketch

> Our understanding of the world is built up of innumerable layers. Each layer is worth exploring, as long as we do not forget that it is one of many. Knowing all there is to know about one layer—a most unlikely event—would not teach us much about the rest.
>
> E. Chargaff, Heraclitean Fire: Sketches from a Life before Nature, 1978

Part III is built around the intelligent process, which is outlined in sections 10–13. Sections 8, 9 are supporting sections.

Thus, in this part, our focus is on various *constructive processes*, mainly those related to the (temporal) *construction of transformations on the basis of the input structs*. Such tranformations either expand the existing class supertransforms or initiate new ones. This explains the change of emphasis in this part.

8 Applicability and appearance of transformations

In this section, we collected two important concepts related to "structural matching" that are also used in the description of the intelligent process.

The following definition encapsulates the idea of an extruct that has *just appeared* in a given struct, i.e. the latest primitive in a given struct (from the viewpoint of the temporal construction of the struct) has *just* enabled the identification of some relabeling of the extruct (see Fig. 21).

Definition 25. For an extruct ε and a struct σ,

$$\sigma \quad = \quad [\pi_1 \dashv \pi_2 \dashv \cdots \dashv \pi_t],$$

we say that **extruct** ε **has just appeared in struct** σ if there exists σ-extruct $\varepsilon_\sigma = \langle \sigma, \mathrm{Iface}\,, \mathfrak{E} \rangle$, where $\varepsilon_\sigma \in \varepsilon$, such that

$$\pi_t \text{ is present in the last 4-tuple of } \mathfrak{E}$$

(see Def. 10). We denote this relation between extruct and struct as follows

$$\varepsilon \lessdot \sigma.$$

►

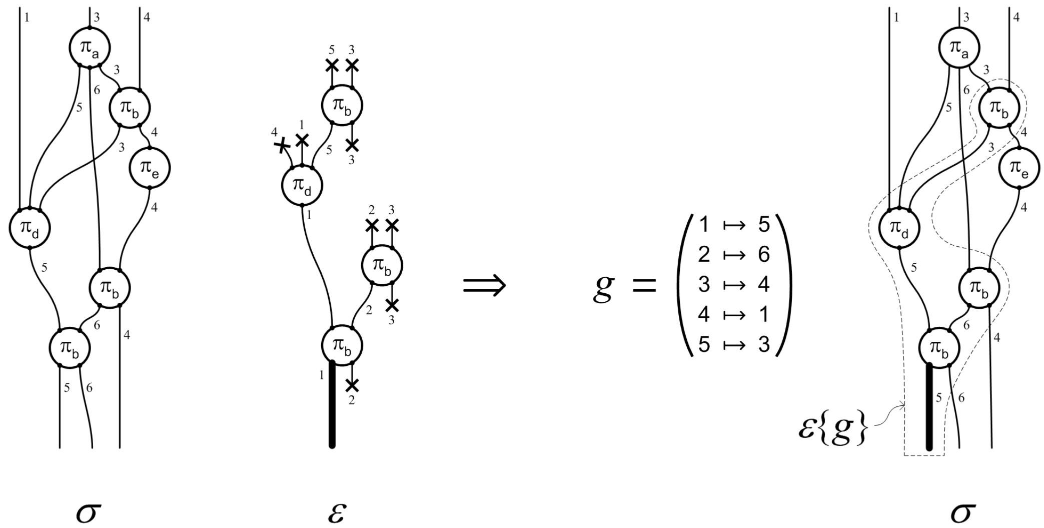


Figure 21: Pictorial illustration of an extruct ε that has just appeared in struct σ: $\varepsilon\{g\} \lessdot \sigma$.

The next concept encapsulates the idea of the "applicability" of a supertransform to a struct.

__Definition__ **26.** If, for some supertransform $\boldsymbol{\tau} = \langle E, B \rangle$ and struct σ, there exists context $\varepsilon \in E$ and extruct site relabeling $g : \mathrm{Sites}\,(\varepsilon) \to SL$ such that

$$\varepsilon\{g\} \lessdot \sigma,$$

then we say that $\boldsymbol{\tau}$ **is currently applicable to struct** σ.

For a given multi-level inductive structure $\mathbb{MIS}$ and a struct $\sigma^{(k)}$ from $\mathbb{MIS}(k) = \langle \mathring{\Pi}^{(k)}, \mathbf{TS}^{(k)} \rangle$, $0 \le k \le l-1$, we define the **set of all k-th level canonical supertransforms that are currently applicable to struct** $\sigma^{(k)}$ as

$$\boldsymbol{T}_{\mathrm{appl}}(\sigma^{(k)}) \ \stackrel{\mathrm{def}}{=}\ \{\mathring{\boldsymbol{\tau}} \in [\boldsymbol{\tau}] \in \mathbf{TS}^{(k)} \mid \mathring{\boldsymbol{\tau}} \text{ is currently applicable to struct } \sigma^{(k)}\}$$

(see Notational Convention 2 after Def. 24). ▶

In a manner similar to the appearance of an extruct, we now introduce the appearance of a transformation (w.r.t. some struct).

__Definition__ **27.** For a transform $\tau = \langle \varepsilon, \beta \rangle$ and a struct σ, we say that **a transform** τ **has just appeared in struct** σ if there exist structs α, γ such that

$$\sigma = \gamma \triangleleft \alpha, \qquad \beta \text{ is a latest substruct of } \alpha \text{ (Def. 6)}, \qquad \varepsilon \lessdot \gamma$$

(see Fig. 22). We denote this relation between transform and struct as follows

$$\tau \lessdot \sigma.$$

▶

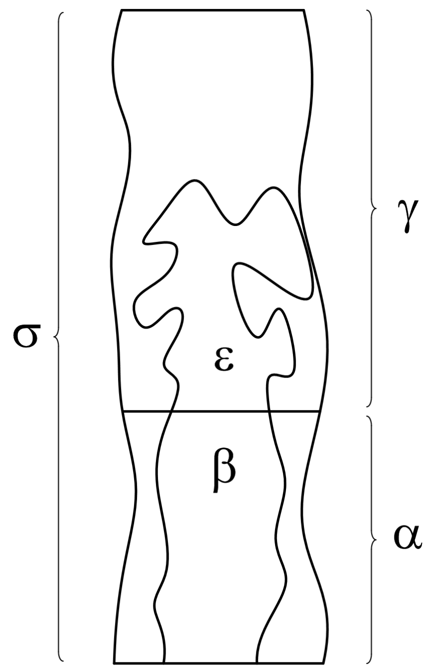


Figure 22: A transform $\tau = \langle \varepsilon, \beta \rangle$ that has just appeared in struct σ (see Def. 27 for notation).

9 Numeric association schemes

In this section, we present a *provisional* (numeric) specification of the structural associations between primitives, between bodies and contexts, and between primitives and structs. At present, we have no choice but to use numeric "weighting" schemes rather than develop structural means for encapsulating the nature of these associations. We do expect, however, that the emergence of relevant applications and hardware will facilitate the development of an appropriate formal language.

Notational convention 3. In what follows, the level index (k) is often dropped for simplicity, when it is clear from the context.

Definition 28. Using the notation of Def. 24, for a k-th representation level $\mathbb{MIS}(k) = \langle \mathring{\Pi}^{(k)}, \mathbf{TS}^{(k)} \rangle$ of a multi-level inductive structure $\mathbb{MIS}$, a **context-body association strength scheme** for class supertransform $[\boldsymbol{\tau}]$, $[\boldsymbol{\tau}] \in \mathbf{TS}^{(k)}$, is defined as a mapping

$$\mathcal{CB}_{\mathring{\boldsymbol{\tau}}} : \{\tau \mid \tau \in \mathring{\boldsymbol{\tau}}\} \to \mathbb{R}_+$$

where $\mathring{\boldsymbol{\tau}} = \langle E, B \rangle$ (see Def. 19). ►

We are now ready to introduce the third fundamental (and most central) concept of the ETS framework.

<u>Definition</u> 29. For a k-th representation level, an (inductive) **class representation** is defined as the following pair

$$\mathcal{CLASS}_{[\boldsymbol{\tau}]} \stackrel{\text{def}}{=} \left\langle\, [\boldsymbol{\tau}]\,,\ \mathcal{CB}_{\mathring{\mathcal{T}}}\,\right\rangle,$$

where class supertransform $[\boldsymbol{\tau}] \in \mathbf{TS}^{(k)}$ and $\mathcal{CB}_{\mathring{\mathcal{T}}}$ is the context-body association strength scheme for $[\boldsymbol{\tau}]$, also called the **class weight scheme**. ▶

In the ETS formalism, the intelligent process relies on class representations to recognize (and also *generate*) objects from a class. The (scalar valued) strength of evidence for the class can be derived from the class weight scheme $\mathcal{CB}_{\mathring{\mathcal{T}}}$.

Notational convention 4. To simplify some formulas in the rest of the paper, we use the notation $\mathring{\pi}(\underline{i})$, $\mathring{\pi}(\overline{j})$ (see Def. 2) to denote not only the corresponding sites, but the following pairs, $\langle\mathring{\pi}, \mathring{\pi}(\underline{i})\rangle$, $\langle\mathring{\pi}, \mathring{\pi}(\overline{j})\rangle$, respectively.

The next definition encapsulates the concept of a (given) weighting scheme specifying the *propensity* for sites to be attached, i.e. the weight between two sites of primitives reflects the observed frequency of their attachment, and thus the projected likelihood of such attachments in the future. To do this, we need to introduce the set of admissible site pairs (see Fig. 23), which will be used in the rest of the section as the "standard" domain for mappings.

<u>Definition</u> 30. For a k-th representation level $\mathbb{MIS}(k)$, with its set of original primitives $\mathring{\Pi}^{(k)} = [\mathring{\pi}_1^{(k)}, \mathring{\pi}_2^{(k)}, \cdots, \mathring{\pi}_{n_k}^{(k)}]$, and its site type mapping TYPE (Def. 21), a **set** LEGAL **of admissible site pairs** (site pair attachments) is

$$\text{LEGAL} \stackrel{\text{def}}{=} \left\{\left\langle \mathring{\pi}_p^{(k)}(\underline{i}), \mathring{\pi}_q^{(k)}(\overline{j})\right\rangle \mid \text{TYPE}\left(\mathring{\pi}_p^{(k)}(\underline{i})\right) = \text{TYPE}\left(\mathring{\pi}_q^{(k)}(\overline{j})\right)\right\}.$$

A (site-wise primitive) attachment strength scheme or **primitive association strength scheme** (PASS) related to the above representation level is a *normalized* mapping

$$\mathcal{AS} : \text{LEGAL} \to \mathbb{R}_+$$

i.e.

$$\sum_{\text{LEGAL}} \mathcal{AS}\left(\, \mathring{\pi}_p^{(k)}(\underline{i}),\, \mathring{\pi}_q^{(k)}(\overline{j})\,\right) \;=\; 1.$$

▶

A useful notation: $[\![\ \ ]\!]$ **.** If some real-valued mapping $\mathcal{A}$ defined on the above set LEGAL

$$\mathcal{A} : \text{LEGAL} \to \mathbb{R}_+$$

is not normalized, then one can easily normalize it to obtain the normalized mapping $[\![\mathcal{A}]\!]$.

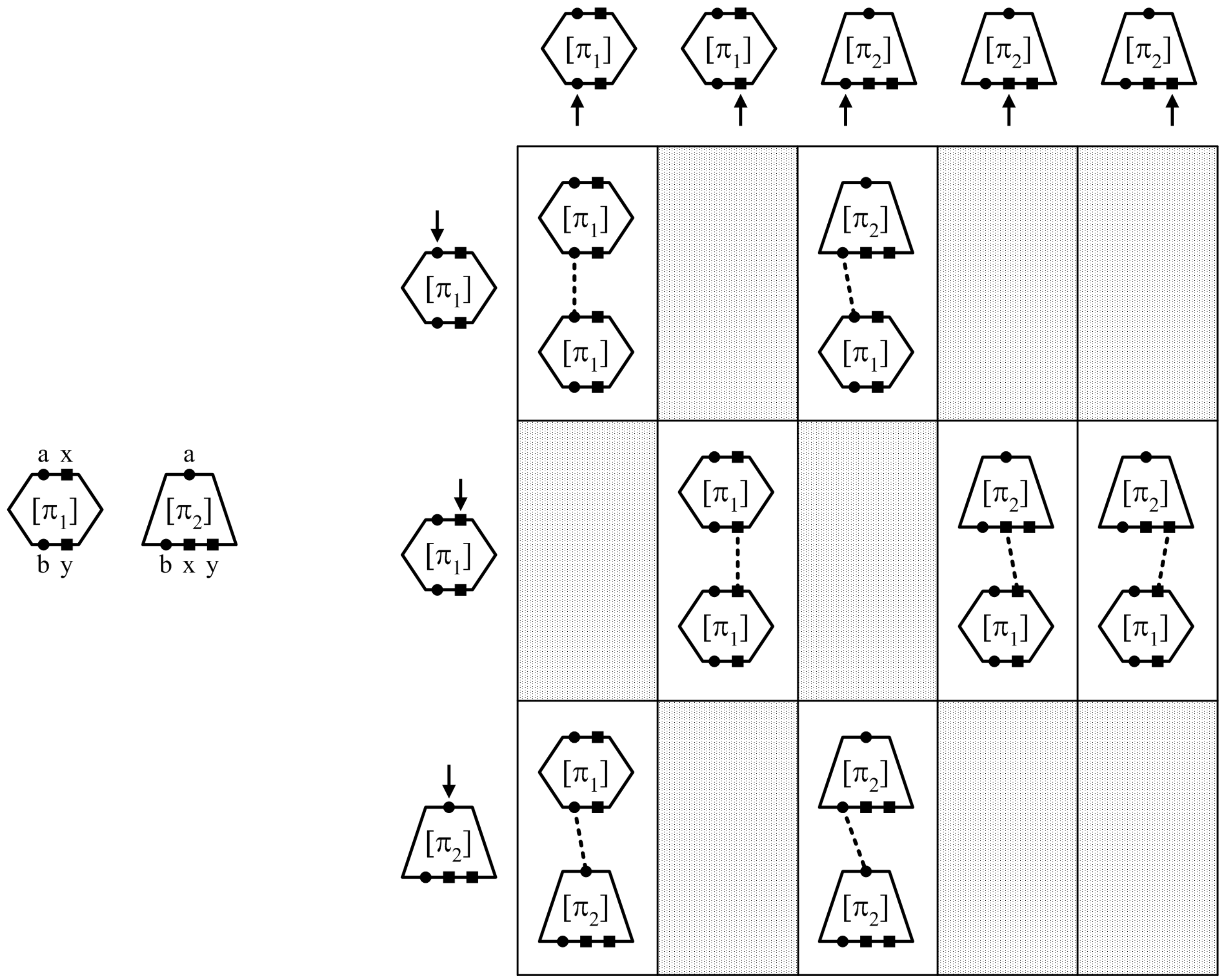


Figure 23: *Left:* two class primitives from Fig. 5. *Right:* a pictorial encapsulation of the corresponding set LEGAL. Each row and each column corresponds to the indicated site of the corresponding original primitive.

The following definition, based on the above PASS, introduces the concept of the propensity of a given primitive to continue a given struct. Moreover, for the rest of the section, recall from Def. 4 that if $\pi_1^{(k)}, \pi_2^{(k)} \in \Pi^{(k)}$, then there exist indices p, q and original primitive site relabelings f_1, f_2 such that

$$\pi_1^{(k)} = \mathring{\pi}_p^{(k)}{}_{\{f_1\}} \qquad\qquad \pi_2^{(k)} = \mathring{\pi}_q^{(k)}{}_{\{f_2\}}.$$

<u>Definition</u> 31. For a fixed representation level (k), let a struct σ be given as

$$\sigma \;=\; [\mathring{\pi}_{n_1}{}_{\{f_1\}} \dashv \mathring{\pi}_{n_2}{}_{\{f_2\}} \dashv \cdots \dashv \mathring{\pi}_{n_t}{}_{\{f_t\}}]$$

(see Lemma 1). Given a primitive π $(= \mathring{\pi}_m{}_{\{f\}})$, the set $\mathrm{LEGAL}\,(\sigma \dashv \pi)$, $\mathrm{LEGAL}\,(\sigma \dashv \pi) \subseteq \mathrm{LEGAL}$, is defined as

$$\mathrm{LEGAL}\,(\sigma \dashv \pi) \;\stackrel{\mathrm{def}}{=}\; \left\{ \langle \mathring{\pi}_{n_p}(\underline{i}), \mathring{\pi}_m(\overline{j}) \rangle \mid \mathring{\pi}_m{}_{\{f\}}(\overline{j}) \text{ is attached to } \mathring{\pi}_{n_p}{}_{\{f_p\}}(\underline{i}) \text{ in } \sigma \dashv \pi \right\}.$$

Then the **continuation strength scheme for** $\sigma \dashv \pi$ is defined as the following restriction of the above primitive association strength scheme $\mathcal{AS}$:

$$\mathcal{AS}[\sigma \dashv \pi] \stackrel{\text{def}}{=} \mathcal{AS}\,|_{\text{LEGAL}\,(\sigma\,\dashv\,\pi)}\,.$$

Moreover, the **strength of the continuation of** σ **by primitive** π is defined as

$$\text{STRN}(\sigma \dashv \pi, \mathcal{AS}) \stackrel{\text{def}}{=} \sum_{x \in \text{LEGAL}\,(\sigma\,\dashv\,\pi)} \mathcal{AS}(x)$$

(see Fig. 24). ►

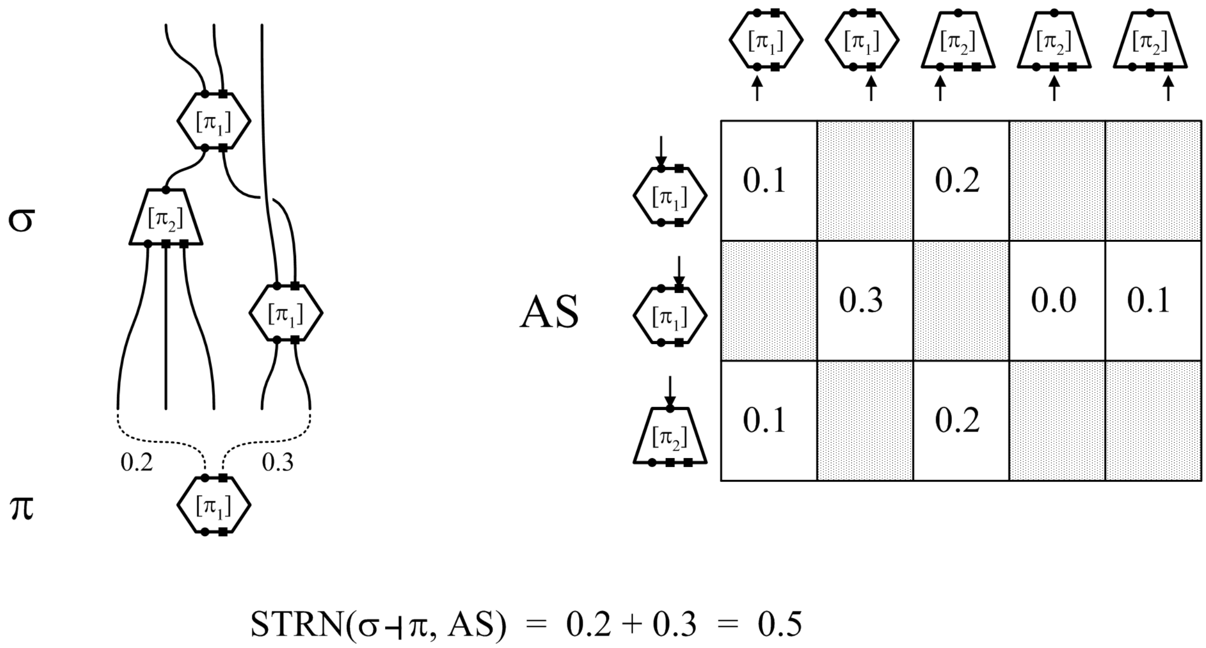


Figure 24: Computation of the strength of the continuation of σ by π for the shown struct σ, primitive π, and set LEGAL shown in Fig. 23.

For the above normalized mappings (see Def. 30), we need the concept of their weighted sum and difference.

<u>Definition</u> 32. For a k-th representation level, let $\mathcal{L}_1, \mathcal{L}_2$ be two normalized mappings, $\mathcal{L}_1, \mathcal{L}_2 : \text{LEGAL} \to \mathbb{R}_+$, and let c_1, c_2 be non-negative reals such that $c_1 + c_2 > 0$. Then the normalized mapping $\mathcal{L} : \text{LEGAL} \to \mathbb{R}_+$, defined as

$$\mathcal{L} \stackrel{\text{def}}{=} [\![\, c_1\,\mathcal{L}_1 + c_2\,\mathcal{L}_2 \,]\!],$$

is called the **weighted sum** of the two normalized mappings, denoted

$$\mathcal{L} \;=\; c_1\,\mathcal{L}_1 \dot{+} c_2\,\mathcal{L}_2\,.$$

Moreover, if $\mathcal{L}_1 = c_2\,\mathcal{L}_2 \dotplus c_3\,\mathcal{L}_3$, then we call $\mathcal{L}_3$ the **weighted difference** of $\mathcal{L}_1$ and $\mathcal{L}_2$, denoted

$$\mathcal{L}_3 \;=\; c_1\,\mathcal{L}_1 \dot{-} c_2\,\mathcal{L}_2 \qquad\qquad (\, c_1 = c_2 + c_3 \,).$$

▶

The following definition introduces integer valued indicator mappings on LEGAL that record simple connectivity information for the context and body of a given transform, i.e. they count the number of times a particular LEGAL attachment has occurred (see Fig. 25).

<u>**Definition**</u> **33.** For a fixed representation level (k), for the above primitives[30] $\pi_1 = \mathring{\pi}_p\,\{f_1\}$, $\pi_2 = \mathring{\pi}_q\,\{f_2\}$ (when considered as constituents of structs corresponding to contexts and bodies), let mapping $I_{\pi_1,\pi_2} : \text{LEGAL} \to \{0, 1\}$ be the following indicator mapping[31]

$$I_{\pi_1,\pi_2}\big(\, \mathring{\pi}_p(\underline{i}),\, \mathring{\pi}_q(\overline{j})\,\big) \;\stackrel{\text{def}}{=}\; \begin{cases} 1 & \text{if } \pi_2(\overline{j}) \text{ is attached to } \pi_1(\underline{i}), \\ 0 & \text{otherwise.} \end{cases}$$

For a transformation $\tau = \langle \varepsilon, \beta \rangle$

$$\varepsilon \;=\; \langle \text{Iface}\,, \mathfrak{E} \rangle,$$

the indicator mapping $I_{\text{cntx}(\tau)} : \text{LEGAL} \to \mathbb{Z}_+$ is defined as[32]

$$I_{\text{cntx}(\tau)} \;\stackrel{\text{def}}{=}\; \sum_{\pi_1,\pi_2 \text{ present in } \mathfrak{E}} I_{\pi_1,\pi_2}\,.$$

Further, if $\alpha \prec \beta$, (Def. 7) then the indicator mapping $I_{\text{body}(\tau)} : \text{LEGAL} \to \mathbb{Z}_+$ is defined as

$$I_{\text{body}(\tau)} \;\stackrel{\text{def}}{=}\; \sum_{\substack{\pi_1 \text{ present in } \mathfrak{E} \text{ or in } \beta \\ \text{and } \pi_2 \text{ present in } \beta}} I_{\pi_1,\pi_2}\,.$$

In other words, $I_{\text{cntx}(\tau)}$ counts the number of times a particular LEGAL attachment has occurred in β or from a primitive in β to a primitive in ε. ▶

The corollary of the next lemma is used in Step 1 of Def. 40.

Lemma 5. For a fixed representation level, given a struct σ and primitive association strength scheme $\mathcal{AS}$, let $\Sigma(\sigma)$ denote a set of valid continuations of σ by various π's:

$$\Sigma(\sigma) \;\stackrel{\text{def}}{=}\; \{\sigma \dashv \pi \mid \pi \in \Pi\}.$$

---

[30] See primitives $\pi_1^{(k)}, \pi_2^{(k)}$ on p. 43.

[31] The first condition is equivalent to: "if the j-th initial site of $\mathring{\pi}_q\,\{f_1\}$ is attached to the i-th terminal site of $\mathring{\pi}_p\,\{f_2\}$".

[32] For simplicity, the common argument of the functions is omitted.

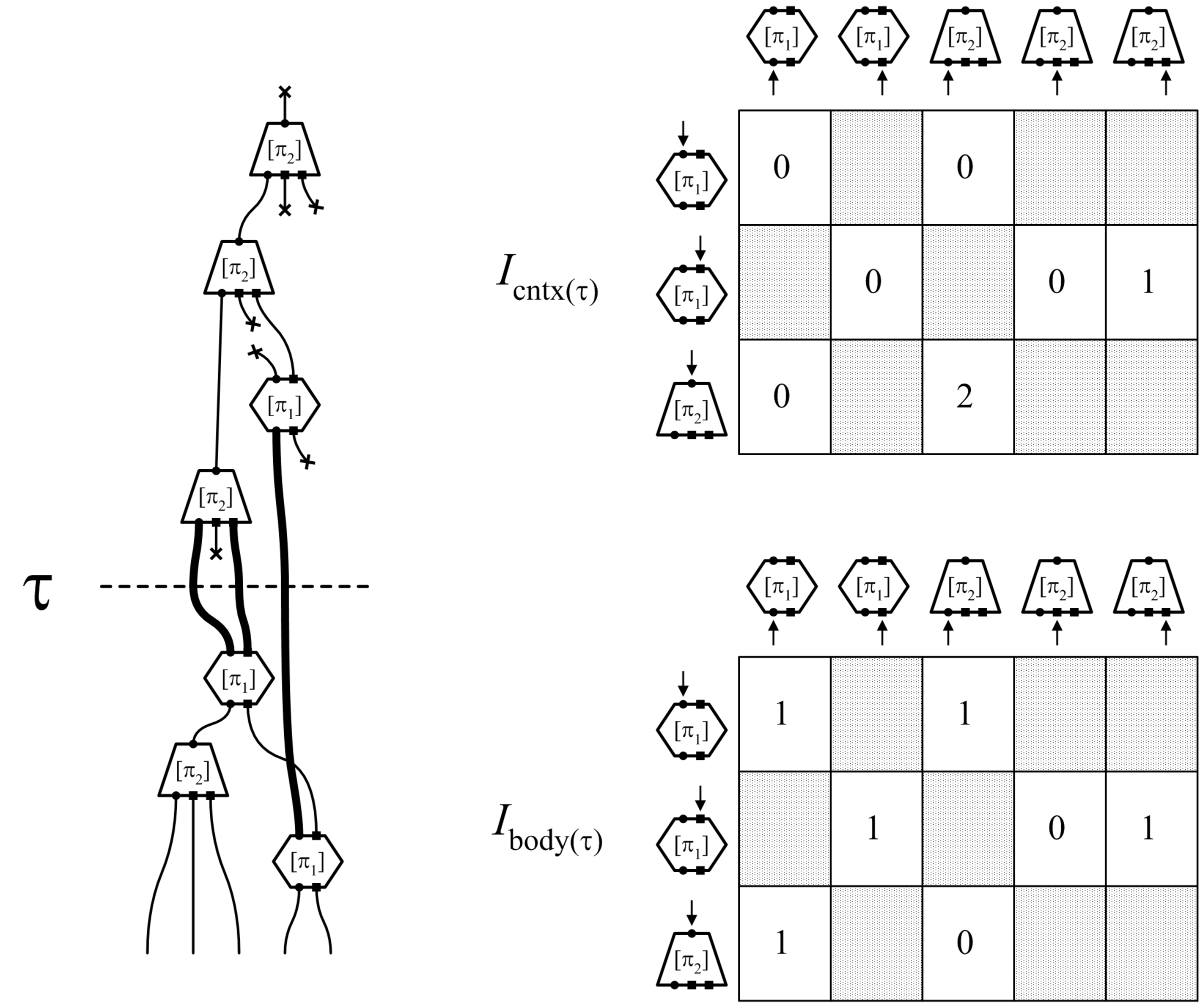


Figure 25: Depiction of mappings $I_{\mathrm{cntx}(\tau)}$ and $I_{\mathrm{body}(\tau)}$ for the shown transform τ.

Moreover, let $\Sigma_i(\sigma)$ be an equivalence class associated with the following equivalence relation on $\Sigma(\sigma)$:

$$\sigma \dashv \pi_1 \cong \sigma \dashv \pi_2 \qquad \Longleftrightarrow \qquad \sigma \dashv \pi_2 \;=\; \sigma \dashv \pi_{1\{f\}}$$

$$\text{for some site relabeling } f \text{ such that } \quad f|_{\mathrm{Sites}\,(\sigma)\,\cap\,\mathrm{Sites}\,(\pi_1)} = \mathrm{id}\ .$$

(Thus $\Sigma(\sigma) = \bigcup\limits_i \Sigma_i(\sigma)$). Then, $\forall\, i$,

$$\forall\ \ \sigma \dashv \pi_1,\ \sigma \dashv \pi_2 \in \Sigma_i(\sigma) \qquad\qquad \mathcal{AS}[\sigma \dashv \pi_1] \;=\; \mathcal{AS}[\sigma \dashv \pi_2].$$

■

Corollary 1. In the notation of the lemma, $\forall\, i$

$$\forall\ \ \sigma \dashv \pi_1,\ \sigma \dashv \pi_2 \in \Sigma_i(\sigma) \qquad\qquad \mathrm{STRN}(\sigma \dashv \pi_1, \mathcal{AS}) \;=\; \mathrm{STRN}(\sigma \dashv \pi_2, \mathcal{AS}).$$

(Thus, it is meaningful to write $\mathrm{STRN}\big(\Sigma_i(\sigma), \mathcal{AS}\big)$). ■

10 The intelligent process and its states: introduction

In this section, we initiate the description of a more appropriate, global, view (associated with the functioning of the intelligent process) of *all* the relevant "construction processes" mentioned previously, e.g. struct construction, extruct construction.

In this paper we address only the *unsupervised* version of the intelligent process, which could be modified to handle the supervised case.

Explanatory definition of the intelligent process. An **intelligent process** $\mathscr{P}$ is a process which "optimally" captures/represents the development[33] of some universe by expanding and refining (in a discrete mode) its multi-level inductive structure $\mathbb{MIS}$, including the number of its levels. It accomplishes this mainly by the creation and modification (but never deletion) of relevant class supertransforms at the appropriate levels.

Sections 12 and 13 contain the relevant formal details related to the description of the intelligent process.

Auxiliary notation. To specify the evolution of process $\mathscr{P}$, we call an instantaneous description of such a process its (hidden) state, denoted **state**, and assume that, associated with each such state of process $\mathscr{P}$, is a multi-level inductive structure $\mathbb{MIS}^{\mathscr{P}}_{\mathbf{state}}$ encapsulating its structural description. Some hidden states are also called observable states, denoted **STATE**.

The following definition introduces the concept of a k-th level in the state of an intelligent process. This results in a multi-level state concept.

<u>**Definition**</u> **34.** For an intelligent process $\mathscr{P}$ with its evolving multi-level inductive structure $\mathbb{MIS}^{\mathscr{P}}$, **the k-th level of its** (hidden) **state** is a 3-tuple

$$\mathbf{state}_{\mathscr{P}}(k) \stackrel{\text{def}}{=} \left\langle\ \mathbb{MIS}^{\mathscr{P}}_{\text{state}}(k),\ \mathbb{WM}^{\mathscr{P}}_{\text{state}}(k),\ \mathbb{AM}^{\mathscr{P}}_{\text{state}}(k)\ \right\rangle,$$

where[34]:

- $\mathbb{MIS}^{\mathscr{P}}_{\text{state}}(k)$ is the corresponding k-th level (single-level) inductive structure (see Def. 24)
- the **k-th level working** (structural) **memory** $\mathbb{WM}^{\mathscr{P}}_{\text{state}}(k)$ is a pair

  $$\mathbb{WM}^{\mathscr{P}}_{\text{state}}(k) \stackrel{\text{def}}{=} \left\langle\ \sigma^{(k)}_{\text{state}},\ USED^{(k)}_{\text{state}}\ \right\rangle,$$

  where $\sigma^{(k)}_{\text{state}}$ is a struct,

  $$\sigma^{(k)}_{\text{state}} = [\pi_1 \dashv \pi_2 \dashv \cdots \dashv \pi_t],$$

[33] More specifcally, we mean an event view of this development, see Sec. 2 and Fig. 3.

[34] The index $\mathscr{P}$ will often be dropped.

called the **working struct**, that records the event history as it is seen from this level and $USED^{(k)}_{\text{state}}$ is a mapping,

$$USED^{(k)}_{\text{state}} : \{1, 2, \ldots, t\} \to \{\text{yes}, \text{no}\},$$

which marks each primitive π_i, $1 \le i \le t$, of $\sigma^{(k)}_{\text{state}}$ as "processed" (or "used") or not[35]; note that for $\sigma^{(k)}_{\text{state}} = \theta$, $USED^{(k)}_{\text{state}}$ does not need to be specified

- the k**-th level association** (schemes) **memory** $\mathbb{AM}^{\mathscr{P}}_{\text{state}}(k)$ is a pair

$$\mathbb{AM}^{\mathscr{P}}_{\text{state}}(k) \overset{\text{def}}{=} \left\langle\, \mathcal{AS}^{(k)}_{\text{state}},\ \{\mathcal{CB}_{\text{state}\,,\,\mathring{\boldsymbol{\tau}}} \mid \mathring{\boldsymbol{\tau}} \in \mathbf{TS}^{(k)}_{\text{state}}\}\, \right\rangle,$$

where mapping $\mathcal{AS}^{(k)}_{\text{state}}$ is used by the process to store recently-observed site-wise primitive attachment strengths (Def. 30) and each mapping $\mathcal{CB}_{\text{state}\,,\,\mathring{\boldsymbol{\tau}}}$ is used to maintain the corresponding class weight schemes (Def. 28).

▶

Note that the working struct $\sigma^{(k)}_{\text{state}}$ can, in fact, be comprised of several "disjoint" structs (i.e. those with no site in common). Moreover, from an applied perspective, it is sufficient to deal only with small working structs[36].

Remark 6. From a conventional point of view, it is useful to think that the "recognition" of a supertransform at level k is signified by the attachment of the corresponding $(k+1)$-level primitive to $\sigma^{(k+1)}_{\text{state}}$. This remark should provide some bridge between conventional PR ideas and those presented here.

The next concept puts all the levels of a state together, and also clarifies the transition between observable states.

<u>Definition</u> 35. The **state of process** $\mathscr{P}$ is defined as

$$\mathbf{state}_{\mathscr{P}} \overset{\text{def}}{=} \left\langle \mathbf{state}_{\mathscr{P}}(0), \mathbf{state}_{\mathscr{P}}(1), \cdots, \mathbf{state}_{\mathscr{P}}(l-1) \right\rangle,$$

where l is the number of levels in $\mathbb{MIS}^{\mathscr{P}}_{\textbf{state}}$.

A **state transition** in an intelligent process $\mathscr{P}$ from some presently observable state $\mathbf{STATE}_{\text{pres}}$ to the **next** observable state $\mathbf{STATE}_{\text{next}}$ is comprised of a sequence of hidden states:

$$\mathbf{state}_{\text{pres}},\ \mathbf{state}_{\text{pres}+1},\ \cdots,\ \mathbf{state}_{\text{last}},$$

where

$$\mathbf{state}_{\text{pres}} \overset{\text{def}}{=} \mathbf{STATE}_{\text{pres}}, \qquad \mathbf{state}_{\text{last}} \overset{\text{def}}{=} \mathbf{STATE}_{\text{next}}.$$

▶

[35] This primitive marking is necessary mainly for the reason that transform bodies are only searched for in the "unprocessed part" of $\sigma^{(k)}_{\text{state}}$.

[36] This is true in view of the observation that, at each level, only a relatively recent event history needs to be considered.

11 Availability of a transformation and fitness of its parts

As was stated in the explanatory definition of the intelligent process, the intelligent process constantly expands and refines its multi-level inductive structure. *In part*, this is accomplished by either creating a new class supertransform (and, therefore, a new primitive) or by expanding an existing one. To this end, the process will first need to identify relevant *candidate* transformations, which are introduced in this section as transforms that have "just become available".

The first definition introduces two numeric measures related to the fitness of each part of a transform, which will be used for the identification of the candidate transforms introduced in the next (main) definition of the section.

<u>**Definition**</u> **36.** For a k-th representation level in state $\mathbf{state}_{\mathscr{P}}$, given two application-dependent constants $d^{(k)}_{\text{cntx}}$ and $d^{(k)}_{\text{body}}$ and a transform $\tau = \langle \varepsilon, \beta \rangle$, $\tau \in \boldsymbol{\tau} \in \mathbf{TS}^{(k)}_{\text{state}}$, the **fitness of the context of** τ is

$$\mathrm{Fit}_{\mathrm{cntx}(\tau)} \stackrel{\text{def}}{=} d^{(k)}_{\text{cntx}} - D\left(\mathcal{AS}^{(k)}_{\text{state}} \dot{-} C^{(k)}_{\tau} \cdot [\![I_{\mathrm{body}(\tau)}]\!] , [\![I_{\mathrm{cntx}(\tau)}]\!] \right) ,$$

where $C^{(k)}_{\tau}$ is the (application-dependent) transform fitness coefficient, and the **fitness of the body of** τ is

$$\mathrm{Fit}_{\mathrm{body}(\tau)} \stackrel{\text{def}}{=} d^{(k)}_{\text{body}} - D\left(\mathcal{AS}^{(k)}_{\text{state}} , [\![I_{\mathrm{body}(\tau)}]\!] \right) ,$$

where D is some fixed dissimilarity/distance measure[37] on the set of all normalized mappings whose domain is LEGAL and $[\![I]\!]$ stands for the normalized mapping obtained by normalizing mapping I (see Def. 30). ▶

<u>**Definition**</u> **37.** We say that a **transformation**[38] $\tau^{(k)} = \langle \varepsilon, \beta \rangle$ **has just become available** (in $\mathbf{state}_{\mathscr{P}}(k)$) of an intelligent process $\mathscr{P}$ if

- *(τ has just appeared in $\sigma^{(k)}_{\text{state}}$)*

$$\tau^{(k)} \lessdot \sigma^{(k)}_{\text{state}} \qquad \text{(Def. 27)}$$

- *(each primitive of* body(τ) *that appears, possibly relabeled, in* $\sigma^{(k)}_{\text{state}}$ *has not been used)*

  for every primitive $\pi^{(k)}$ in β $\qquad USED^{(k)}_{\text{state}}(j) = \text{no}$

  where j is the index of the primitive in $\sigma^{(k)}_{\text{state}}$ corresponding to the relabeled $\pi^{(k)}$

[37] E.g. the most popular, Euclidean, distance measure.

[38] This transformation may not be in any of the class supertransforms from $\mathbf{TS}^{(k)}_{\text{state}}$.

- *(the next-level primitive induced by τ is attachable to $\sigma^{(k+1)}_{\text{state}}$)*
  if, for a supertransform $\mathring{\boldsymbol{\tau}}^{(k)} = \{\tau\}$, formed by the single transform τ, the next-level continuation
$$\sigma^{(k+1)}_{\text{state}} \dashv \mathring{\pi}^{(k+1)}$$
$$\text{where } \mathring{\boldsymbol{\tau}}^{(k)} \longrightarrow \mathring{\pi}^{(k+1)},$$
  is legal

- *(τ is "fit" with respect to $\mathcal{AS}^{(k)}_{\text{state}}$)*
$$\text{Fit}_{\text{cntx}(\tau)} > 0 \qquad\qquad \text{Fit}_{\text{body}(\tau)} > 0.$$

For the process $\mathscr{P}$ in state $\textbf{state}_{\mathscr{P}}(k)$, the set $T_{\text{avail}}\big(\textbf{state}_{\mathscr{P}}(k)\big)$ is defined as **the set of all transforms that have just become available in** $\textbf{state}_{\mathscr{P}}(k)$. ▶

12 Process initiation and state transitions

We now introduce the very first state of the intelligent process.

Definition 38. The **initial** (observable) **state**, $\textbf{STATE}_{\text{initial}}$, **of an intelligent process** $\mathscr{P}$ is any valid state of process $\mathscr{P}$ (see Def. 35), and *possibly* even with the following "null state":
$$\textbf{STATE}_{\text{initial}} \stackrel{\text{def}}{=} \big\langle\, \big\langle\, \mathbb{MIS}_{\text{initial}}(0),\ \mathbb{WM}_{\text{initial}}(0),\ \mathbb{AM}_{\text{initial}}(0)\,\big\rangle\,\big\rangle,$$
where
$$\begin{aligned} \mathbb{MIS}_{\text{initial}}(0) &\stackrel{\text{def}}{=} \big\langle\, \mathring{\Pi}^{(0)}_{\text{initial}}\,,\ \varnothing\,\big\rangle \\ \mathbb{WM}_{\text{initial}}(0) &\stackrel{\text{def}}{=} \big\langle\, \theta\,,\ USED^{(0)}_{\text{initial}} \ \text{ is an empty mapping}\,\big\rangle \\ \mathbb{AM}_{\text{initial}}(0) &\stackrel{\text{def}}{=} \big\langle\, \mathcal{AS}^{(0)}_{\text{initial}}\,,\ \varnothing\,\big\rangle. \end{aligned}$$
▶

We note that the intelligent process never halts, in view of the nature of the transition initiation stage (see Step 1 in Def. 40).

Next, we introduce the basic organization of a transition between *any* two consecutive observable states.

Definition 39. We divide an observable state transition from the present observable state $\textbf{STATE}_{\text{pres}}$ $(=\textbf{state}_{\text{pres}})$ to the next observable state $\textbf{STATE}_{\text{next}}$ $(=\textbf{state}_{\text{last}})$[39], into two stages: the **transition initiation stage** and the **multi-level transition stage**. ▶

[39] See Def. 35.

The first of the latter two stages is relatively simple and basically consists of choosing a primitive at *level zero* in order to continue the level zero working struct. However, before proceeding with the definition of the transition initiation stage, we need to briefly mention the role of "sensor(s)" in the intelligent process. Without going into detail, we assume the process relies on its sensor(s), whose main function is related to the simultaneous updating of the association strength schemes $\mathcal{AS}^{(k)}$ at a number of levels s, where s depends on the structure of the sensor.

Definition 40. The **transition initiation stage** is comprised of the following three steps.

Step 1 (parallel update of association memory by sensor(s)). Sensor(s) update association strength schemes at several levels by updating the entries associated with sensed data.

Step 2 (primitive selection at level zero). Let

$$\mathbb{WM}_{\text{pres}}(0) \;=\; \big\langle \sigma^{(0)}_{\text{pres}}, USED^{(0)}_{\text{pres}} \big\rangle,$$

where

$$\sigma^{(0)}_{\text{pres}} \;=\; [\pi^{(0)}_1 \dashv \pi^{(0)}_2 \dashv \cdots \dashv \pi^{(0)}_t].$$

The choice of the primitive $\pi^{(0)}_{\text{pres}}$ $(= \mathring{\pi}^{(0)}_{\text{pres}\,\{f\}})$ to be attached to the working struct is performed by, first, randomly choosing[40] $\Sigma_i(\sigma^{(0)}_{\text{pres}})$ (from the corresponding partition of $\Sigma(\sigma^{(0)}_{\text{pres}})$) with probability proportional to $\text{STRN}\big(\Sigma_i(\sigma^{(0)}_{\text{pres}}), \mathcal{AS}^{(0)}\big)$, and, second, by selecting a particular $\sigma^{(0)}_{\text{pres}} \dashv \pi^{(0)}_{\text{pres}}$ from the chosen $\Sigma_i(\sigma^{(0)}_{\text{pres}})$.

Step 3 (advance to the next state by updating working memory at level zero). Define the next hidden state as

$$\textbf{state}_{\text{pres}+1} \overset{\text{def}}{=} \big\langle \textbf{state}_{\text{pres}+1}(0),\ \textbf{state}_{\text{pres}}(1), \cdots, \textbf{state}_{\text{pres}}(l-1) \big\rangle,$$

in which the only level being updated is level 0

$$\textbf{state}_{\text{pres}+1}(0) \;\overset{\text{def}}{=}\; \langle\, \mathbb{MIS}_{\text{pres}}(0),\, \mathbb{WM}_{\text{pres}+1}(0),\, \mathbb{AM}_{\text{pres}+1}(0) \,\rangle,$$

where the last two entries are defined as follows:

- *(continue the working struct by* $\pi^{(0)}_{\text{pres}}$ *)*

$$\mathbb{WM}_{\text{pres}+1}(0) \;\overset{\text{def}}{=}\; \big\langle\, \sigma^{(0)}_{\text{pres}} \dashv \pi^{(0)}_{\text{pres}}\,,\ USED^{(0)}_{\text{pres}+1} \,\big\rangle,$$

$USED^{(0)}_{\text{pres}+1}$ is an extension of $USED^{(0)}_{\text{pres}}$ with $USED^{(0)}_{\text{pres}+1}(t+1) = \text{no}$

[40] For the notation, see Lemma 5 and its corollary in Sec. 9. This complication is an artifact of the necessity of dealing with a possibly infinite set of relabeled primitives.

- *(increase association strengths between $\pi^{(0)}_{\text{pres}}$ and the primitives it was just attached to)*

$$\mathbb{AM}_{\text{pres}+1}(0) \overset{\text{def}}{=} \left\langle\ \mathcal{AS}^{(0)}_{\text{pres}+1}\,,\ \{\mathcal{CB}_{\text{pres}}\,,\mathring{\tau} \mid \mathring{\boldsymbol{\tau}} \in \mathbf{TS}^{(0)}_{\text{pres}}\}\ \right\rangle,$$

$$\mathcal{AS}^{(0)}_{\text{pres}+1} = \mathcal{AS}^{(0)}_{\text{pres}}\ \dot{+}\ \left[\!\!\left[\mathcal{AS}^{(0)}_{\text{pres}}\,[\sigma^{(0)}_{\text{pres}} \dashv \pi^{(0)}_{\text{pres}}]\right]\!\!\right]\ .$$

▶

We are now ready to present the organization of the *main* stage in the transition between two observable states.

<u>**Definition**</u> **41.** The **multi-level transition stage**, at *each* level k (beginning with $k = 0$), is comprised of the following three consecutive steps[41], comprising a **k-th level transition substage**:

- the **learning step** (Def. 42)
- the **recognition step** (Def. 43)
- the **facilitation step** (Def. 44).

On completion of the k-th level transition substage, check whether the **termination flag**[42] has been raised: if so, then the entire multi-level transition stage is completed, otherwise proceed to the next level, level $(k + 1)$.

A more detailed depiction of the multi-level transition stage is presented in Fig. 26. We assume that the multi-level transition stage terminates at some level m, where $0 \leq m \leq l-1$.
▶

We note that, at present, the names of the steps express their *intended* meaning, rather than the current, possibly inadequate, implementations of the corresponding steps.

13 The k-th level transition substage

This section is written in an algorithmic style, which is more appropriate, given the nature of the subject matter, and it's "definitions" are actually algorithmic descriptions of the corresponding steps. For implementational details, see [10].

We strongly suggest that, during the reading of this section, readers should convince themselves that, as expected, structural considerations transcend numeric ones in the functioning of the intelligent process. Since all definitions in this section refer to the same level k, we will often omit the level index (k) unless confusion arises. We suggest that it may be useful to review carefully Fig. 26. Also note the following convention when presenting each of the following steps: in each case, *after* updating the start state $\mathbf{state}_{\text{step-start}}$, we switch to the notation $\mathbf{state}_{\text{step}}$ to denote the resulting *hidden* state.

[41] Each of these steps involves a single hidden state transition.
[42] The termination flag may be implemented as a boolean variable set to "FALSE" at the beginning of *each* transition initiation stage (Def. 40).

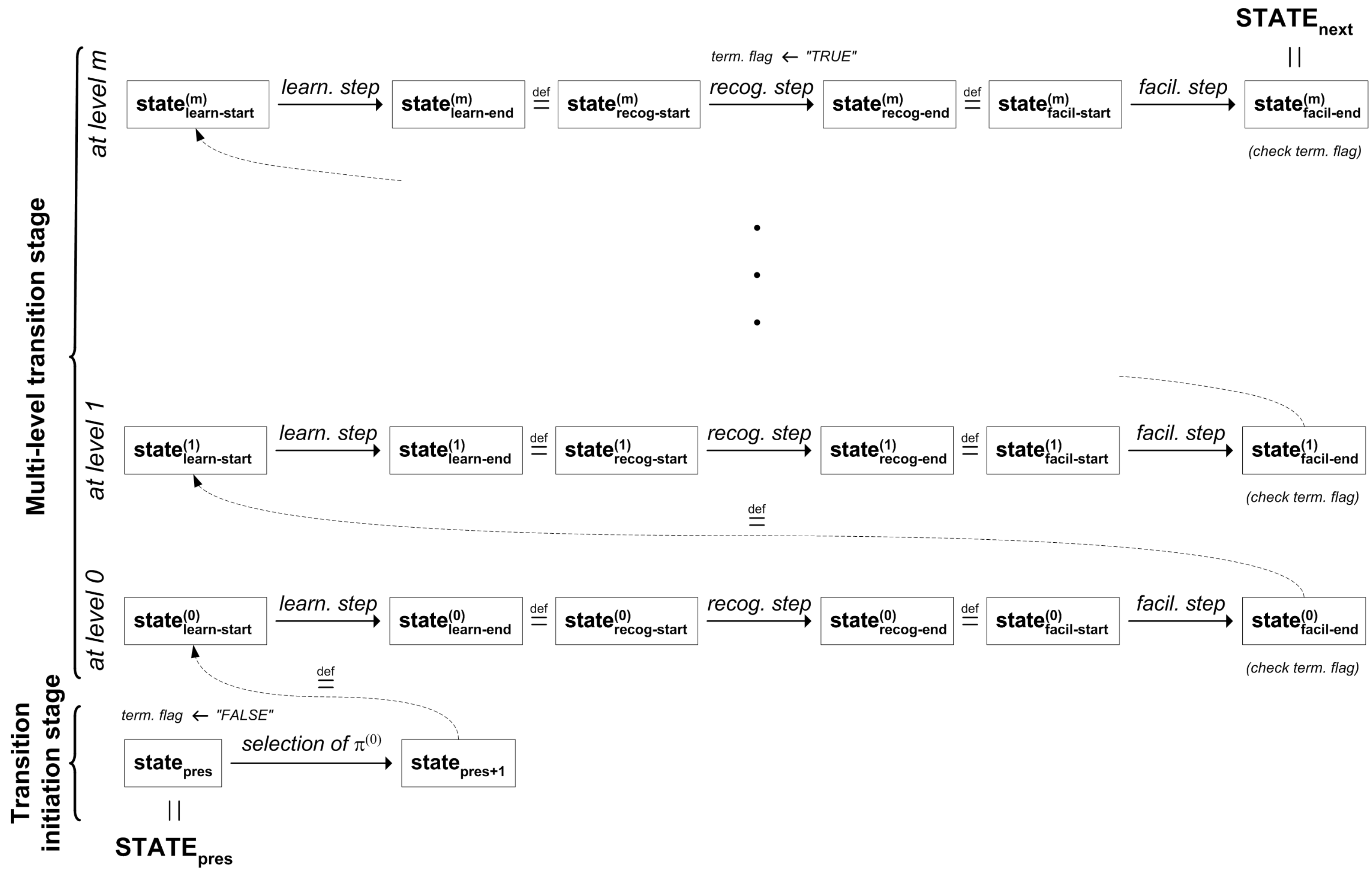


Figure 26: *Notational convention* for the multi-level transition stage. The recognition and learning steps involve substeps (not shown).

13.1 The learning step

This subsection outlines the first (learning) step in the k-th level transition substage, which involves the possible discovery of new or modification of existing class descriptions at this level.

Definition 42. For a k-th level transition substage of an intelligent process $\mathscr{P}$, the **learning step** consists of the following 3 substeps.

Substep 1. Identify the set $T^{(k)}_{\text{avail}}(\,\textbf{state}_{\text{learn-start}}\,)$ of all transforms that have just become available in $\textbf{state}_{\text{learn-start}}$ (see Def. 37). If $T^{(k)}_{\text{avail}}(\,\textbf{state}_{\text{learn-start}}\,) = \varnothing$ then end the learning step, i.e. $\textbf{state}_{\text{learn-end}} = \textbf{state}_{\text{learn-start}}$.

Substep 2. Choose a transform υ from $T^{(k)}_{\text{avail}}(\,\textbf{state}_{\text{learn-start}}\,)$ for which $\text{Fit}^{(k)}_{\text{body}(\upsilon)}$ is maximal (with ties broken based on $\text{Fit}^{(k)}_{\text{cntx}(\upsilon)}$).

Substep 3. For the chosen transform υ,

$$\upsilon \;=\; \langle \varepsilon, \beta \rangle,$$

- *(υ has been learned previously)*
  IF $\upsilon \in \boldsymbol{\tau} \in \textbf{TS}^{(k)}_{\text{learn-start}}$, then end the learning step

- *(β has been learned previously)*
  IF, for some[43] class supertransform $[\boldsymbol{\tau}] \in \textbf{TS}^{(k)}_{\text{learn-start}}$ (with canonical supertransform $\mathring{\boldsymbol{\tau}}$), $[\boldsymbol{\tau}, \text{cntx}(\upsilon)]$ is a valid context expansion of $[\boldsymbol{\tau}]$ w.r.t. υ (Def. 20), then update $\textbf{state}_{\text{learn-start}}$ as follows:

  – update $\mathbb{AM}_{\text{learn-start}}(k)$ by replacing only one of its class weight schemes, i.e. $\mathcal{CB}_{\text{learn-start}\,,\,\mathring{\boldsymbol{\tau}}}$ (for the above $[\boldsymbol{\tau}]$), with its extension $\overline{\mathcal{CB}}_{\text{learn-start}\,,\,\mathring{\boldsymbol{\tau}}_1}$ (corresponding to $[\boldsymbol{\tau}, \text{cntx}(\upsilon)]$ whose canonical supertransform is $\mathring{\boldsymbol{\tau}}_1$)

$$\overline{\mathcal{CB}}_{\text{learn-start}\,,\,\mathring{\boldsymbol{\tau}}_1}(\tau) \;\stackrel{\text{def}}{=}\; \begin{cases} \mathcal{CB}_{\text{learn-start}\,,\,\mathring{\boldsymbol{\tau}}}(\tau) & \tau \in \mathring{\boldsymbol{\tau}} \\ \sum\limits_{x \,\in\, \text{LEGAL}} I_{\text{body}(\tau)}(x) & \exists\, h \quad \tau = \upsilon\{h\} \\ \text{implementation-dependent} & \text{otherwise} \end{cases}$$

  – replace $[\boldsymbol{\tau}]$ with $[\boldsymbol{\tau}, \text{cntx}(\upsilon)]$

- *(ε has been learned previously)*
  IF, for some class supertransform $[\boldsymbol{\tau}] \in \textbf{TS}^{(k)}_{\text{learn-start}}$, $[\boldsymbol{\tau}, \text{body}(\upsilon)]$ is a valid body expansion of $[\boldsymbol{\tau}]$ w.r.t. υ, then update $\textbf{state}_{\text{learn-start}}$ as follows:

[43] Note that there exists at most one such class supertransform $[\boldsymbol{\tau}]$.

– update $\mathbb{AM}_{\text{learn-start}}(k)$ as above, exchanging the roles of the context and body

– replace $[\boldsymbol{\tau}]$ with $[\boldsymbol{\tau}, \text{body}(\upsilon)]$

- *(create a new k-th level class representation $\mathcal{CLASS}^{(k)}_{[\boldsymbol{\tau}]}$, Def. 29, and in the case $k = l - 1$, a new level l)*
  ELSE

  – create a new supertransform $\mathring{\boldsymbol{\tau}}^{(k)}_{\text{new}} = \{\upsilon\}$, i.e. $\mathring{\boldsymbol{\tau}}^{(k)}_{\text{new}} = \langle E_{\text{new}}, B_{\text{new}} \rangle$, where $E_{\text{new}} = \{\varepsilon\}$ and $B_{\text{new}} = \{\beta\}$

  – if $k < l - 1$

  * update $\mathbb{MIS}_{\text{learn-start}}(k)$ as follows:

$$\mathbf{TS}^{(k)}_{\text{learn}} = \mathbf{TS}^{(k)}_{\text{learn-start}} \cup \{[\mathring{\boldsymbol{\tau}}^{(k)}_{\text{new}}]\}$$

  * update $\mathbb{MIS}_{\text{learn-start}}(k+1)$ as follows:

$$\mathring{\Pi}^{(k+1)}_{\text{learn}} = \mathring{\Pi}^{(k+1)}_{\text{learn-start}} \cup \mathring{\pi}^{(k+1)}_{\text{new}}, \qquad \mathring{\boldsymbol{\tau}}^{(k)}_{\text{new}} \rightarrow \mathring{\pi}^{(k+1)}_{\text{new}}$$

  – if $k = l - 1$,[44]

  * update $\mathbb{MIS}_{\text{learn-start}}(k)$ as follows:

$$\mathbf{TS}^{(k)}_{\text{learn}} = \{[\mathring{\boldsymbol{\tau}}^{(k)}_{\text{new}}]\}$$

  * create a new level $l = k + 1$ (see Def. 34):

$$\begin{aligned} \mathbb{MIS}_{\text{learn}}(l) &\overset{\text{def}}{=} \left\langle \{\mathring{\pi}^{(l)}_{\text{new}}\}, \varnothing \right\rangle \\ \mathbb{WM}_{\text{learn}}(l) &\overset{\text{def}}{=} \left\langle \theta, \text{empty mapping} \right\rangle \\ \mathbb{AM}_{\text{learn}}(l) &\overset{\text{def}}{=} \left\langle \mathcal{AS}^{(l)}_{\text{new}}, \varnothing \right\rangle \end{aligned}$$

  where the values of $\mathcal{AS}^{(l)}_{\text{new}}$ are defined in an application-dependent manner.

  – update $\mathbb{AM}_{\text{learn-start}}(k)$ as follows: add to the set of class weight schemes the new class weight scheme defined for the above class supertransform $\mathring{\boldsymbol{\tau}}^{(k)}_{\text{new}}$ (consisting of the single transform υ) $\mathcal{CB}_{\mathring{\boldsymbol{\tau}}^{(k)}_{\text{new}}} : \{\upsilon\} \rightarrow \mathbb{R}_+$

$$\mathcal{CB}_{\mathring{\boldsymbol{\tau}}^{(k)}_{\text{new}}}(\upsilon) \overset{\text{def}}{=} \sum_{x \in \text{LEGAL}} I_{\text{body}(\upsilon)}(x).$$

►

Note that, when a new level l is created (at the end of substep 3), the number of levels in the corresponding $\mathbb{MIS}$ is increased by one. The number of levels changes nowhere else in the intelligent process.

[44] Recall that, for this k, $\mathbf{TS}^{(k)}_{\text{learn-start}} = \varnothing$ (Def. 24).

13.2 The recognition step

We now outline the next (recognition) step in the k-th level transition substage. This step involves, first, the recognition of some transform τ $(\tau \in \boldsymbol{\tau} \in \mathbf{TS}^{(k)}_{\text{recog-start}}$, see notational convention 2 on p. 37) that has just become available (see Def. 37) and, second, the consequent updates: the continuation of the *next-level* working struct $\sigma^{(k+1)}_{\text{recog-start}}$ by the next-level primitive corresponding to $\boldsymbol{\tau}$, as well as updates of $\mathbb{AM}_{\text{recog-start}}(k)$, $\mathbb{AM}_{\text{recog-start}}(k+1)$, and $\mathbb{WM}_{\text{recog-start}}(k)$.

Definition 43. For a k-th level transition substage of an intelligent process $\mathscr{P}$, the **recognition step** consists of the following 4 substeps.

Substep 1. Identify the set $T^{(k)}_{\text{avail}}(\,\mathbf{state}_{\text{recog-start}}\,)$ of all transforms that have just become available in $\mathbf{state}^{(k)}_{\text{recog-start}}$ (see Def. 37), further denoted $T^{(k)}_{\text{avail}}$. Next, select a subset $\overline{T}^{(k)}_{\text{avail}}$ of the latter set of those transforms that are also in some class supertransform from $\mathbf{TS}^{(k)}_{\text{recog-start}}$:

$$\overline{T}^{(k)}_{\text{avail}} \stackrel{\text{def}}{=} \left\{\tau \in T^{(k)}_{\text{avail}} \mid \exists\, \boldsymbol{\tau} \in \mathbf{TS}^{(k)}_{\text{recog-start}} \text{ such that } \tau \in \boldsymbol{\tau}\right\}.$$

If the last set is empty (which is always the case at the last level), then raise the termination flag (Def. 41) and skip substeps 2 and 3.

Substep 2. Randomly choose a transform $\tau^{(k)}$ (and corresponding $\pi^{(k+1)}$) to be "recognized" from set $\overline{T}^{(k)}_{\text{avail}}$ with probability proportional to $\mathrm{STRN}\left(\sigma^{(k+1)}_{\text{recog-start}} \dashv \pi^{(k+1)}\,,\, \mathcal{AS}^{(k+1)}_{\text{recog-start}}\right)$.

Substep 3. Since the above transform τ is in $T^{(k)}_{\text{avail}}$, τ must have just appeared in working struct $\sigma^{(k)}_{\text{recog-start}}$: $\tau \lessdot \sigma^{(k)}_{\text{recog-start}}$ (Def. 37). Let us denote by β its relabeled body:

$$\beta \stackrel{\text{def}}{=} \mathrm{body}(\tau)_{\{g\}}.$$

Update $\mathbf{state}_{\text{recog-start}}$ as follows, where $\pi^{(k+1)}$ is from substep 2:

- *(continue the next-level working struct by* $\pi^{(k+1)}$ *)*
  update $\mathbb{WM}_{\text{recog-start}}(k+1)$:

$$\sigma^{(k+1)}_{\text{recog-end}} = \sigma^{(k+1)}_{\text{recog-start}} \dashv \pi^{(k+1)}$$

- *(increase association strengths between* $\pi^{(k+1)}$ *and the primitives it was just attached to)*
  update the PASS in $\mathbb{AM}_{\text{recog-start}}(k+1)$:

$$\mathcal{AS}^{(k+1)}_{\text{recog-end}} = \mathcal{AS}^{(k+1)}_{\text{recog-start}} \dot{+} \left[\!\left[\mathcal{AS}^{(k+1)}_{\text{recog-start}}\left[\sigma^{(k+1)}_{\text{recog-start}} \dashv \pi^{(k+1)}\right]\right]\!\right]$$

- *(mark as processed all primitives of* $\sigma^{(k)}_{\text{recog-start}}$ *that appear, possibly relabeled, in body* β *)*
  update $\mathbb{WM}_{\text{recog-start}}(k)$:

$$\forall\, \pi \in \beta \qquad\qquad USED^{(k)} \text{ of the corresponding index } = \text{ yes}\,.$$

<u>Substep 4.</u> Update $\mathbb{AM}_{\text{recog-start}}(k)$ as follows:

- *(for each known transform that has just become available, decrease the association strengths between the primitives in its body)*
  update the current PASS:

$$\mathcal{AS}^{(k)}_{\text{recog-end}} \;=\; \mathcal{AS}^{(k)}_{\text{recog-start}} \;\dot{-}\; C^{(k)}_{\text{recog}} \cdot \left[\!\!\left[\sum_{\tau \in \overline{T}^{(k)}_{\text{avail}}} \text{Fit}_{\text{body}(\tau)} \cdot I_{\text{body}(\tau)} \right]\!\!\right],$$

  where $C^{(k)}_{\text{recog}}$ is the (application-dependent) transform decrement coefficient

- *(for each previously-learned transform that has just become available, increase its context-body association strength)*
  update $\left\{ \mathcal{CB}_{\text{recog}\,,\,\mathring{\boldsymbol{\tau}}} \mid \mathring{\boldsymbol{\tau}} \in \mathbf{TS}^{(k)}_{\text{recog-start}} \right\}$:

  for each $\tau \in \overline{T}^{(k)}_{\text{avail}}$ and site relabeling $h : \text{Sites}\,(\tau) \to SL$ such that $\tau\{h\} \in \mathring{\boldsymbol{\tau}}$, do

$$\mathcal{CB}_{\text{recog}\,,\,\mathring{\boldsymbol{\tau}}}(\tau\{h\}) \;=\; \mathcal{CB}_{\text{recog}\,,\,\mathring{\boldsymbol{\tau}}}(\tau\{h\}) + \text{Fit}_{\text{body}(\tau)},$$

  leaving $\mathcal{CB}_{\text{recog}\,,\,\mathring{\boldsymbol{\tau}}_i}$ unchanged for the remaining $\mathring{\boldsymbol{\tau}}_i \in \mathbf{TS}^{(k)}_{\text{recog-start}}$.

▶

Note that, in the last part of substep 4, since several τ's may belong to the same relabeled $\mathring{\boldsymbol{\tau}}$, then (the same) $\mathcal{CB}_{\mathring{\boldsymbol{\tau}}}$ may be updated repeatedly.

13.3 The facilitation step

Finally, we outline the third and last (facilitation) step in the k-th level transition substage. This step facilitates (but does not guarantee) the future appearance of certain transformations by raising their "status" (via $\mathcal{AS}^{(k)}$). A review of Defs. 31, 33 is recommended since substantial *numeric* calculations are concentrated in this step.

<u>Definition</u> 44. For a k-th transition substage of an intelligent process $\mathscr{P}$, the **facilitation step** consists of the following 4 substeps.

Substep 1. For the current working struct $\sigma^{(k)}_{\text{facil-start}}$, identify the set of all k-th level canonical supertransforms that are currently applicable to it as $\boldsymbol{T}_{\text{appl}}(\sigma^{(k)}_{\text{facil-start}})$, which we denote $\boldsymbol{T}^{(k)}_{\text{appl}}$ (Def. 26).

Substep 2. For each canonical supertransform $\mathring{\boldsymbol{\tau}}$ from $\boldsymbol{T}^{(k)}_{\text{appl}}$, define an indicator mapping $I_{\mathring{\boldsymbol{\tau}}}$

$$I_{\mathring{\boldsymbol{\tau}}} \stackrel{\text{def}}{=} \sum_{\substack{\text{over all } \tau \in \mathring{\boldsymbol{\tau}} \text{ such that } \exists g \\ \text{cntx}(\tau)\{g\} \,\lessdot\, \sigma^{(k)}_{\text{facil-start}}}} \mathcal{CB}_{\text{facil-start}\,,\mathring{\boldsymbol{\tau}}}(\tau) \cdot I_{\text{body}(\tau)}.$$

Substep 3. For the current PASS, compute the increment corresponding to the above indicator mappings, where, for each $\mathring{\boldsymbol{\tau}}^{(k)}$, $g^{\star}$ is any legitimate extension to set $\text{Sites}\,(\mathring{\boldsymbol{\tau}}^{(k)})$ of one of the corresponding site relabelings g [45] from substep 2:

$$\text{Incr}(\mathcal{AS}^{(k)}_{\text{facil-start}}) \stackrel{\text{def}}{=} \left[\!\!\left[\sum_{\substack{\mathring{\boldsymbol{\tau}}^{(k)} \in \boldsymbol{T}^{(k)}_{\text{appl}} \\ \mathring{\pi}^{(k+1)} \to \mathring{\boldsymbol{\tau}}^{(k)}}} \left[\text{STRN}\left(\sigma^{(k+1)}_{\text{facil-start}} \dashv \mathring{\pi}^{(k+1)}\{\overline{g^{\star}}\}\,,\, \mathcal{AS}^{(k+1)}_{\text{facil-start}}\right)\right] \cdot I_{\mathring{\boldsymbol{\tau}}} \right]\!\!\right],$$

$$\overline{g^{\star}} = g^{\star}|_{\text{Sites}\,(\mathring{\pi}^{(k+1)})}.$$

Note that the corresponding *next-level* continuation strengths are used as coefficients in this formula.

Substep 4. Facilitate the future appearance of transforms from the above supertransforms by updating the current PASS in $\mathbb{AM}_{\text{facil-start}}(k)$:

$$\mathcal{AS}^{(k)}_{\text{facil-end}} = \mathcal{AS}^{(k)}_{\text{facil-start}} \dot{+} C^{(k)}_{\text{facil}} \cdot \text{Incr}(\mathcal{AS}^{(k)}_{\text{facil-start}}),$$

where $C^{(k)}_{\text{facil}}$ is the (application-dependent) transform increment coefficient. ▶

Part IV
Conclusion

14 High-priority directions

- Development of applications (and supporting *representations*) in biology, bioinformatics, cheminformatics, various pattern recognition areas, data mining, information retrieval, etc.

[45] For each $\mathring{\boldsymbol{\tau}}$, the choice of the particular g is irrelevant *if* the values of the extension on $\text{Sites}\,(\mathring{\boldsymbol{\tau}}) \setminus \text{Sites}\,(\text{cntx}(\tau))$ are fixed, since the restriction of $g^{\star}$ appearing in the formula is not affected by these values.

- Development of new hardware for and new algorithmic approaches to structural matching, e.g. finding transforms in structs.
- Development of an intelligent process that includes intelligent subprocesses as modular units.
- Develop a more adequate version of the level ascension postulate.
- Develop improved numeric association schemes, e.g.
  - arrange the decay of $\mathcal{CB}_{\mathring{\mathcal{T}}}$ weights to give more recent transforms a fair chance;
  - $\mathcal{AS}^{(k)}$ weight adjustments during facilitation should be performed in a cascaded/temporal manner, rather than all at once;
  - arrange greater stability of weights at lower levels as compared to higher levels.
- Major revisions of this version of the ETS model, including
  - clarification of the nature of sites in primitives (and thus of primitives themselves);
  - introduction of parallelism in the processing of working structs (including "sensory" data);
  - modification of the current concept of a transform body to a new concept similar to an "inverted" extruct, to support a more dynamic interaction/interplay between transforms in the working structs;
  - generalization of the above concept of a transform to a new concept that looks like an extruct without interface sites, with constituent primitives partitioned into initial, middle, and terminal parts (see section 15.2).

Note that since the next section was added after this section was completed, the above should be modified in light of the ideas presented in section 15.

15 Last-minute brainstorming ideas

After the paper was completed, a number of (potential) significant *simplifications* of basic ETS concepts emerged, some of which are mentioned below.

15.1 Partial order of primitives in structs and a more structural association memory

A modified concept of struct might be defined as a partially ordered set of primitives, as opposed to a linearly ordered set (Fig. 27 (a)). This generalization removes the strong constraint of total linear ordering of primitive events, information which is often unavailable.

From the point of view of the intelligent process, this demands the incorporation of not just a sequential, but also of a parallel mode of processing. This quite happily leads to a

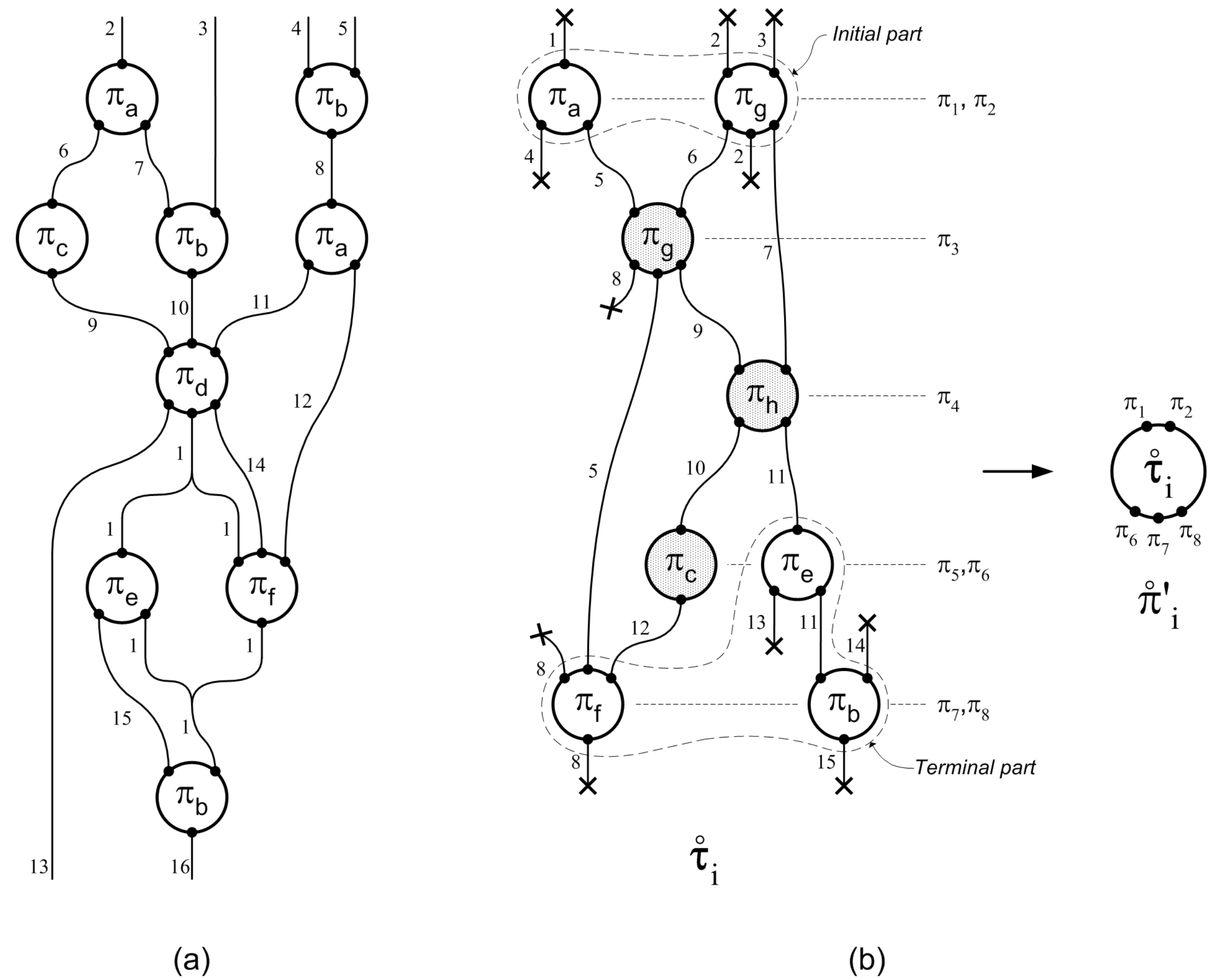


Figure 27: (a) Pictorial illustration of the "new" concept of struct: primitives depicted on the same horizontal line are not linearly ordered w.r.t. each other; two primitives sharing only "through" sites and having no primitives "between" them that force a linear order are depicted as having "split" through sites (π_e, π_f). (b) A transformation whose body is an "inverted extruct" and the next-level primitive corresponding to it.

revision of the procedure (in the intelligent process) for the selection of primitives to be appended to the working struct as well as the procedure for the facilitation of transformations. In particular, in the selection of primitives, the numeric association strength scheme is replaced with a *more appropriate*, *structural*, version of this concept, i.e. by a pool of weighted "primitives-in-waiting", the members of which are used in the construction of various kinds of continuations (parallel or otherwise) of working structs, transformations, etc. In the case of the facilitation procedure, the primitives from the body of a transform (which is being facilitated) are added to the above pool of primitives. The learning procedure is also modified in an appropriate manner.

15.2 The level ascension postulate

First, we assume that a transform now looks like that mentioned at the very end of section 14. In this case, the next-level primitive is constructed as follows: a transform is partitioned into three parts, initial, middle, and terminal, and primitives in the initial and terminal parts are used to "construct" the sites of the next-level primitive (Fig. 27 (b)). Moreover, an equivalence relation between transforms with such partitions ensures that allowable variability is restricted to the middle part.

15.3 An entirely different perspective

In an alternate basic approach, *sites* are "regenerating processes": a site is a set of (interchangeable) process segments, represented by structs that have the same numbers/types of sites on the initial and terminal ends, correspondingly. Such structs can "continue" each other for some period of time, thus capturing the process of regeneration. A primitive transformation, then, represents a change in the character of the intelligent process from the initial site-processes to the terminal site-processes. In other words, a primitive transformation can be thought of as capturing a "standard" change in the pattern of the sites-subprocesses: from its initial subprocesses to its terminal subprocesses.

References

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

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

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

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

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

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

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

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

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

[10] D. Gay, Graduate thesis, Faculty of Computer Science, UNB, in preparation.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] L. Goldfarb, What is a representational formalism?, in preparation.

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

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

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

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

[25] 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.

[26] A. G. Kurosh, *Lectures on General Algebra*, Chelsea, New York, 1963, Section 1.5.

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

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

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

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

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

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

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

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

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

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

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

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

[39] D. Wolpert, Preface in: D. Wolpert (ed.), *The Mathematics of Generalization*, Addison-Wesley, Reading, MA, USA, 1995, p. 8.

[40] 3rd International Workshop on Hybrid Methods for Adaptive Systems, [http://adiret.cs.uni-magdeburg.de/∼nuernb/hmas2003/], Oulu, Finland, July 2003.

[41] 4th International Workshop on Hybrid Methods for Adaptive Systems, [http://adiret.cs.uni-magdeburg.de/hmas2004/], Aachen, Germany, June 2004.

Appendix: Index of main concepts

$\mathring{\pi}$	original primitive	Def. 1 (p. 13)
$\mathring{\pi}(\overline{k})$	k-th initial site of $\mathring{\pi}$	Def. 2 (p. 14)
$\mathring{\pi}(\underline{l})$	l-th terminal site of $\mathring{\pi}$	Def. 2 (p. 14)
$\pi\ (= \mathring{\pi}\{f\})$	primitive	Def. 4 (p. 15)
$[\pi]$	class primitive	Def. 4 (p. 15)
σ	struct	Def. 5 (p. 17)
$\sigma \dashv \pi$	continuation of struct σ by primitive π	Def. 5 (p. 17)
$\alpha \triangleleft \beta$	composition of structs α and β	Def. 7 (p. 21)
$\alpha \prec \beta$	struct β is composable with struct α	Def. 7 (p. 21)
$\sigma\{g\}$	site-relabeled struct σ for site relabeling g	Def. 8 (p. 22)
$\langle \sigma, \mathrm{Iface} \rangle$	interfaced struct with interface sites Iface	Def. 10 (p. 23)
$\mathfrak{e}^i_\sigma$	constituent of the interfaced struct $\langle \sigma, \mathrm{Iface} \rangle$	Def. 10 (p. 23)
ε_σ	σ-extruct	Def. 11 (p. 25)
ε	extruct	Def. 12 (p. 27)
$\varepsilon\{g\}$	site-relabeled extruct ε for site relabeling g	Def. 13 (p. 27)
$\tau = \langle \varepsilon, \beta \rangle$	transformation (or transform)	Def. 14 (p. 28)
$\mathrm{cntx}(\tau)$	context of transform τ	Def. 14 (p. 28)
$\mathrm{body}(\tau)$	body of transform τ	Def. 14 (p. 28)
$\tau\{h\}$	site-relabeled transform τ for site relabeling h	Def. 15 (p. 28)
$\boldsymbol{\tau} = \langle E, B \rangle$	supertransform	Def. 16 (p. 29)
$E \times B$	constituent transform set of supertransform $\boldsymbol{\tau}$	Def. 16 (p. 29)
$[\boldsymbol{\tau}]$	class supertransform for supertransform $\boldsymbol{\tau}$	Def. 18 (p. 30)
$\mathring{\boldsymbol{\tau}} \in [\boldsymbol{\tau}]$	canonical supertransform for $[\boldsymbol{\tau}]$	Def. 19 (p. 31)
$[\boldsymbol{\tau}, \mathrm{cntx}(\upsilon)]$	context expansion of class supertransform $[\boldsymbol{\tau}]$ w.r.t. transform υ	Def. 20 (p. 32)
$[\boldsymbol{\tau}, \mathrm{body}(\upsilon)]$	body expansion of class supertransform $[\boldsymbol{\tau}]$ w.r.t. transform υ	Def. 20 (p. 32)
TS	transformation system	Def. 21 (p. 34)
$\mathring{\pi}'$	level 1 (next to level 0) original primitive	Def. 21 (p. 34)
$\mathbb{MIS}$	multi-level inductive structure	Def. 24 (p. 36)
$\mathbb{MIS}(k)$	k-th level inductive structure	Def. 24 (p. 36)

$\varepsilon \lessdot \sigma$	extruct ε has just appeared in struct σ	Def. 25 (p. 39)
$\boldsymbol{T}_{\text{appl}}(\sigma^{(k)})$	the set of all k-th level canonical supertransforms that are currently applicable to struct $\sigma^{(k)}$	Def. 26 (p. 40)
$\tau \lessdot \sigma$	transform τ has just appeared in struct σ	Def. 27 (p. 40)
$\mathcal{CB}_{\mathring{\boldsymbol{\tau}}}$	context-body association strength scheme for class supertransform $[\boldsymbol{\tau}]$ with canonical supertransform $\mathring{\boldsymbol{\tau}}$	Def. 28 (p. 41)
$\mathcal{CLASS}_{[\boldsymbol{\tau}]}$	class representation for class supertransform $\mathring{\boldsymbol{\tau}}$	Def. 29 (p. 42)
LEGAL	set of admissible site pairs	Def. 30 (p. 42)
$\mathcal{AS}$	primitive association strength scheme (PASS)	Def. 30 (p. 42)
$\mathcal{AS}[\sigma \dashv \pi]$	continuation strength scheme for $\sigma \dashv \pi$	Def. 31 (p. 43)
$\mathrm{STRN}(\sigma \dashv \pi, \mathcal{AS})$	strength of the continuation of struct σ by π	Def. 31 (p. 43)
$[\![\mathcal{A}]\!]$	mapping obtained by normalizing mapping $\mathcal{A}$ defined on set LEGAL	Notation (p. 42)
$c_1\,\mathcal{L}_1 \dot{+} c_2\,\mathcal{L}_2$	weighted sum of normalized mappings $\mathcal{L}_1, \mathcal{L}_2$	Def. 32 (p. 44)
$\mathcal{I}_{\text{cntx}(\tau)}$	indicator mapping for the context of τ	Def. 33 (p. 45)
$\mathcal{I}_{\text{body}(\tau)}$	indicator mapping for the body of τ	Def. 33 (p. 45)
$\mathbf{state}_{\mathscr{P}}(k)$	k-th level state of intelligent process $\mathscr{P}$	Def. 34 (p. 47)
$\mathbb{WM}^{\mathscr{P}}_{\text{state}}(k)$	k-th level working memory of process $\mathscr{P}$	Def. 34 (p. 47)
$\sigma^{(k)}_{\text{state}}$	working struct (in $\mathbb{WM}^{\mathscr{P}}_{\text{state}}(k)$)	Def. 34 (p. 47)
$\mathbb{AM}^{\mathscr{P}}_{\text{state}}(k)$	k-th level association memory of process $\mathscr{P}$	Def. 34 (p. 47)
$\mathbf{state}_{\mathscr{P}}$	state of process $\mathscr{P}$	Def. 35 (p. 48)
$\mathbf{STATE}_{\mathscr{P}}$	observable state of process $\mathscr{P}$	Def. 35 (p. 48)
$\mathrm{Fit}_{\text{cntx}(\tau)}$	fitness of the context of transform τ	Def. 36 (p. 49)
$\mathrm{Fit}_{\text{body}(\tau)}$	fitness of the body of transform τ	Def. 36 (p. 49)
$T^{(k)}_{\text{avail}}\big(\mathbf{state}_{\mathscr{P}}(k)\big)$	the set of all transforms that have just become available in $\mathbf{state}_{\mathscr{P}}(k)$	Def. 37 (p. 49)