# Representational Formalism in Which Syntax and Semantics Are Congruent: Towards the Resolution of Searle's Chinese Room Challenge

**Lev Goldfarb**
goldfarb@unb.ca
Inductive Information Systems, 29 Pembroke Crescent
Fredericton, NB E3B 2V1 Canada

**Abstract**

A recently developed formalism for structural representation, called Evolving Transformations System (ETS)—in which syntax and semantics are congruent—suggests an unforeseen constructive resolution to the ubiquitous syntax/semantics incompatibility issue raised by Searle's Chinese Room argument. The ETS is an outcome of long-term research work directed at the development of a category-oriented formalism for structural object representation, which turned out to be an event-based representation.

**Keywords:** representational formalisms, structural representation; artificial intelligence; cognitive science; categories and concepts; Searle's Chinese Room argument; syntax vs. semantics.

## Introduction: The Challenge of Searle's Chinese Room

In the well-known Chinese Room argument (e.g. Preston & Bishop, 2002) [1], John Searle points out that in any known computational formalism—and, one can add, *in any known spoken language or any formalism in science*—the syntax is not related to the semantics. Moreover, since all present day computers are "purely syntactic machines", he claims, they cannot, in principle, be 'intelligent' in the sense we or other biological species are: they cannot relate to what their programs are executing.

I suggest that his argument raises an *issue critical to the development of artificial intelligence, and cognitive science in general*: What is the nature of biological representation that allows it to 'resolves' the issue of syntax/semantics incompatibility? Although it has not been perceived as such, the problem appears to be truly pivotal to the development of cognitive science (and philosophy), since, as was mentioned above, none of our languages, scientific or non-scientific, have resolved this issue: *we have no example of a language or a formalism in which syntax and semantics are congruent*. (For example, in English, the syntactic structure of 'cat' has nothing to do with the semantic structure of cat.)

I suspect that the reason why this particular representational issue has not received yet the attention it deserves has to do with the latter situation: since, in each known language or formalism, the syntax and semantics are completely independent, *we have become completely habituated to this state of affairs*.

Again, why should the resolution of the above question be considered pivotal to the development of cognitive science? There are, at least, four fundamental reasons. As to the first one[2], John Searle and some discussions of the Chinese Room argument have already justified, though somewhat indirectly, its importance in a reasonably satisfactory manner (Preston & Bishop, 2002). Briefly, the reason has to do with the price one has to pay when relying on any of the known representational formalisms: the price is related to the resulting 'mechanistic', and therefore fundamentally inadequate, cognitive models. In fact, Searle's argument suggests that without the representational formalism in which the syntax/semantics incompatibility is eliminated any substantial progress in cognitive science is simply impossible.

The second reason follows from the first one, as soon as we, having accepted the need for a fundamentally new kind of representational formalism, realize the urgency and the monumental nature of the challenge facing cognitive science in this respect.

Finally and most importantly, the third and the fourth reasons are related to the need to investigate the *plausibility* of such new kinds of representational formalisms and, once such a formalism is discovered, to study and understand the *implications* of an agent functioning on its basis.

## The Importance of Congruence of Syntax and Semantics in a Representational Formalism

Here, I suggest a more direct, 'technical' reason for the necessity of the syntax/semantics congruence by considering some of the consequences of relying on a *representational* formalism that does not satisfy the congruence of syntax and semantics (**CSS**) property. Basically, in such a formalism, the *structure* of syntactic constructions is unrelated to the *structure* of semantic constructs. However, before proceeding with these considerations, we need to clarify, at least to some extent, the meaning of the terms 'syntax' and 'semantics' in our context.

In the context of our discussion, *taking the formal structure of the representational formalism seriously*, it is quite reasonable to assume that the term '**syntax**'—conventionally associated with the study of the structure of sign systems—will, more specifically, refer to the

[1] Some suggested that it "is arguably the twentieth century's greatest philosophical polarizer" (Bringsjord & Noel, 2002), while others (Pat Hayes) proposed to "define cognitive Science as 'the ongoing research program of showing Searle's Chinese Room Argument to be false" (Harnad, 2002).

[2] I consider it in the next section.

*underlying formal structure*[3] of the agent's, or internal, representational formalism. At the same time, the term '**semantics**' will refer to the underlying, or postulated, (external) structure of the agent's environment. Although the latter designation is not standard, I believe it is simpler, more precise, and also quite productive, as will become clear below.

Now, let us assume that we built an agent functioning on the basis of the representational formalism which *does not* satisfy the above CSS property. It is understood, of course, that our agent's architecture must include a 'sensory' mechanism responsible for realizing a mapping $\boldsymbol{f}$ that maps each admissible external object $\boldsymbol{o}$ into its representation $\underline{\boldsymbol{o}}$ in the agent's 'mind' [4]. Next, let us consider some object $\underline{\boldsymbol{o}}^*$ (internally) constructed (*and not directly sensed*) by the agent in accordance with the embedded syntax. What can we say about a *possible* external counterpart $\boldsymbol{o}^*$ of this internally constructed object $\underline{\boldsymbol{o}}^*$ ? The insurmountable difficulties one is facing when trying to address this question are directly related to the postulated structure of the agent's representational formalism. Indeed, the lack of the CSS property does not allow one to produce a *semantically* meaningful 'reverse' mapping [5] $\boldsymbol{g}$, $\boldsymbol{g}$: $\underline{\boldsymbol{o}}^* \mapsto \boldsymbol{o}^*$, since in order to be able to produce such a mapping the (syntactic) structure of the agent's representational formalism must be correlated with the (semantic) structure of its environment. In other words, although the two structures don't have to be *exactly* identical, their underlying formal (axiomatic/computational) structures *must be of the same type*: in the latter case, the attempted above two-way correspondence is possible and meaningful: *on both sides of the divide, the objects are composed in the same manner, prescribed by the common formal, or 'computational', structure*. In fact, below, I outline an example of a representational formalism, the Evolving Transformations System (ETS) formalism (Goldfarb et al., 2007), that embodies, as far as I know, the first example of such formal structure.

One of the main goals of this paper is to motivate the quest for the representational formalisms that satisfy the CSS property.

Incidentally, it is not difficult to see that the above argument also suggests that the syntax of a natural language, at least as it has been understood so far, cannot be indicative of the basic/underlying (biological) representational formalism.

**Numeric Representations** There are a number of reasons for believing that any CSS formalism must be a formalism for *structural*, rather than numeric, object representation. Basically, to produce a numeric representation, of necessity, one must dismantle the original *structural* information[6] in order to encode it numerically, thus leaving the missing (from the vector representation) structural/semantic information to be supplied by the human mind. The same mind, of course, also had to be involved in the original 'dismantling'.

In other words, the semantic information appears to be synonymous with the (properly formalized) structural, or relational, information.

## General Features of the ETS Formalism

Not surprisingly, ETS formalism was not developed in response to the Chinese Room argument: it was motivated by the needs—emerging within the field of pattern recognition in the 60's and 70's—for a *unified* formalism for object representation. Hence, before proceeding with the informal sketch of several of its basic concepts in the next section, it might be useful to get an overall impression of the ETS.

**Background** The formalism began as an attempt to unify two principal approaches to pattern recognition: the vector space approach (in which the pattern classes are delineated by the optimal decision surfaces) and the formal grammars approach (in which a class representation is specified by the formal, or generative, grammar). Each of the two incompatible approaches has its own advantages and disadvantages (Goldfarb, 1990; Goldfarb & Nigam, 1995).

The present version of the ETS (Goldfarb et al., 2007) has gradually evolved out of many demanding attempts to convert the initial ideas in Goldfarb (1990) into an event-based representational formalism, which necessitated radical rethinking of the conventional mathematical language.

**Objects as Structured Processes** First of all, it is important to emphasize that in the ETS formalism, for the first time, *objects are viewed and represented as structured processes* (or *structs*). More accurately, each object is represented as an interconnected (temporal) sequence of structured *primitive events*. Each such primitive event transforms/interrupts the regular flow of several adjacent *primal* processes (of undisclosed structure) specific to this event, and, as a result of the event, transformed primal processes continue to 'flow' until, again, some event intervenes (see Figure 2).

It appears that ETS is the first formalism consistent with the process philosophy (Rescher, 2008) advocated by a

[3] As has been the accepted practice in mathematics during the last half a century, the *generative* 'rules', the rules for constructing (syntactically valid) objects, must rely on the postulated formal/axiomatic structure *only*. E.g. in the case of *the vector space representational formalism*, when representing a handwritten character $\mathcal{B}$ by a vector, one cannot rely on any non-linear—and therefore external to the (linear) axiomatic structure of the vector space—relationship among the coordinates of the vector representing $\mathcal{B}$. Thus, although we can represent various $\mathcal{B}$'s in a vector space, we have no *syntactically valid way* of deciding which of the (arbitrary) chosen vectors represents a character $\mathcal{B}$.

[4] In the ETS formalism, $\boldsymbol{f}$ preserves some additional structure.

[5] The inverse of $\boldsymbol{f}$ may not exist, since $\boldsymbol{f}$ might well be not an injective mapping.

[6] E.g. non-numeric relationships between various part of $\mathcal{B}$.

number of philosophers, most prominently in the last century by Alfred North Whitehead (e.g. Whitehead, 1978).

**Object Classes and Class Representation** The concept of class, or category, pervades all levels of consideration in ETS. For example, primitive events are defined by means of the classes of primal processes; also, one of the very central concepts is that of the class of structs, which relies on the concept of (generative) class representation.

**Representational Stages** Given a particular *stage* of representation, the *next stage* emerges when global patterns of structs, each resembling a magnified primitive event, are each *shrunk* into a new, or the next stage, primitive event (Figure 7). Thus, the introduction of new stages allows one to substantially reduce complexity of the representation.

**Representational Levels** Even *within a single representational stage*, there are hierarchical *levels* of representation associated with the *nested* modular, or compositional, structure of structs, i.e. associated with the nested partitions of a struct (Figure 5).

## Sketch of Basic ETS Concepts

The space limitations allow us to consider only some of the basic concepts and only informally. For a fuller exposition see Goldfarb et al. (2007). Before proceeding, however, it useful to keep in mind that the structure of the ETS formalism has no analogues to compare it with, despite the fact that its main entities, 'structs', may have a superficial resemblance to some other known discrete objects, e.g. graphs.

### Primitive Transformations

The basic ETS concept is that of a primitive event, or primitive transformation, or simply **primitive**, examples of which are depicted in Figure 1.

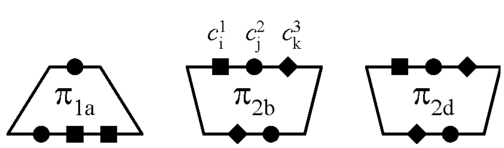


Figure 1: Pictorial depiction of three primitives. The first subscript stands for the class of primitives sharing the same structure, e.g. $\pi_{\mathbf{2b}}$ and $\pi_{\mathbf{2d}}$. An *initial class* of processes is shown as a solid shape on the top, while a *terminal class* is that on the bottom of each event. The only *concrete* processes—i.e. the elements of these classes—labeled in the figure are the initial processes of primitive $\pi_{\mathbf{2b}}$, with label $b = < c_i^1, c_j^2, c_k^3 >$, where $c_t^s$ is the $t^{\text{th}}$ process in primal class $C_s$, $s = 1, 2, 3$.

Although this concept is quite intricate, in another, more important sense, it is simple, since, as will become clear shortly, it carries *identical* semantic and syntactic loads.

Again, a **primitive** stands for a *fixed kind of micro-event* [7] (or process interaction) responsible for transforming **initial processes** (top), into **terminal processes** (bottom); see also Figure 2, where both kinds are shown as lines connecting the primitives. The formal structure of the event is such that it does not depend on the concrete initial (or concrete terminal) processes, as long as each of the processes involved belongs to the corresponding (fixed) class of processes. At this, initial (or $0^{\text{th}}$), stage of representation [8], the structure of each initial and terminal process is suppressed, as is the internal structure of the event itself, and so the formal structure captures the 'external' structure of the event.

Since all of nature is composed of various temporal processes, examples of the above events are all around us: e.g. an elementary particle collision; formation of a two-cell blastula from a single cell (initial process is the original cell and the terminal processes are the resulting two cells); two cars pass by each other; the event associated with the effect on the listener's memory of the sentence "Alice and Bob had a baby" (initial processes are related to Alice and Bob and the terminal processes to Alice, Bob, and the baby).

### Structs

The second basic ETS concept is that of a **struct** [9] formed by a (temporal) sequence of primitives, as shown in Figure 2.

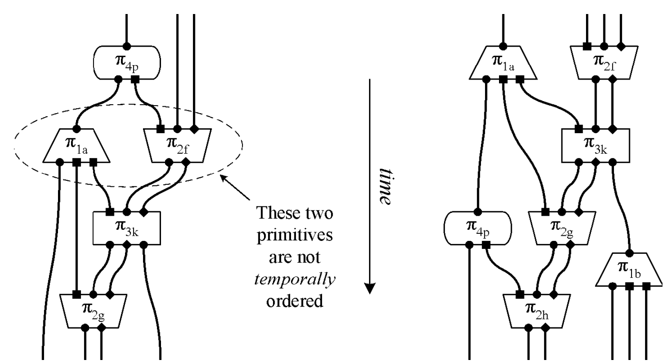


Figure 2: Two structs.

It is easy to see now how the Peano construction of natural numbers (Figure 3) was generalized to the construction of structs: the single 'structureless' unit, out of which a number is built, was replaced by one of several structural ones, i.e. by ETS primitives. An immediate and important consequence of the multiplicity of units is that we can now see *which unit was attached and when*. Hence, the resulting (object) representation, for the first time, embodies both temporal and structural information in the form of a formative, or generative, object history recorded as a series of (structured) events. Consequently, a struct can justifiably

[7] The internal structure of such event is suppressed.

[8] In this paper, I discuss *almost* exclusively a single-stage version of ETS. For multi-stage version, see Goldfarb et al. (2007), Part IV.

[9] Or **level 0 struct** (struct at the initial representational level).

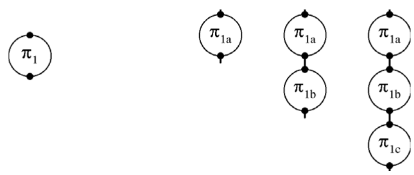

Figure 3: The single primitive involved in the ETS representation of natural numbers (left) and three structs representing the numbers 1, 2, and 3.

be thought of as *the true structural generalization of the numeric representation* (i.e. not reducible to it).

There are two basic contexts within which the above concept of object's formative history can appear. On the agent's side (syntax), a struct is the recorded sequence of sensory micro-events during *the agent's sensory interaction with the target object*. Such interaction must rely, of course, on the agent's own arsenal of primitives. On the objective, agent independent, side (semantics), a struct is the representation of the sequence of events that were actually part of *the object's formation and evolution*. The essential point to observe is that both modes of object representation are captured *naturally* within the same formalism, which appears to be a necessary feature of a representational formalism satisfying the above CSS property.

## Level 0 Classes

The third basic concept is that of a **class**, or category, which can possibly be multi-leveled. A single-level (1-level or level 0) [10] class is defined via a single-level, or level 0, **class generating system**. The latter details the stepwise mode of construction of the class elements. Each (non-deterministic) **step** by such a system—which always follows a (possible) step by the environment—is specified by a *set* of (level 0) constraints. A **constraint**—a non-trivial concept not introduced here, see Goldfarb et al. (2007), Part III—is a formal *specification of a family of structs* sharing structural components in the form of similar substructs. During the above step in the construction process, the struct that is being attached by the system to the part of the class element that has been assembled so far (see Figure 4) *must satisfy one of the constraints specified for this step*.

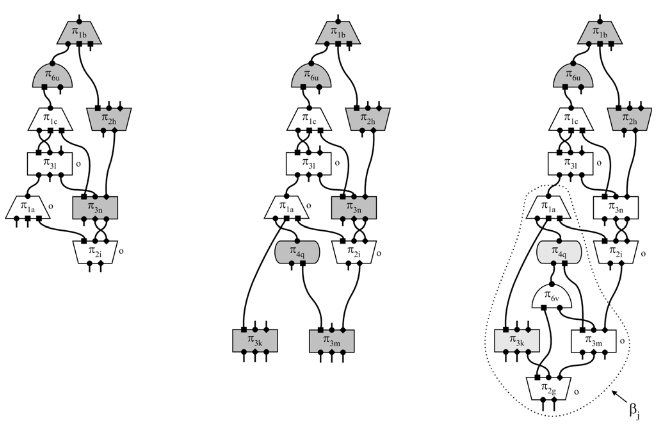

Figure 4: Illustration of a generic *two-step* generative "unit" in the construction of a level 0 class element: a *step by the environment* (bottom three shaded primitives in the second struct) is followed by a *step made by the class generating system* (substruct $\beta_j$ of the third struct, partly overlaying the second struct).

[10] Note the alternative terminology: "single", or "1" *in font of* "level" refer to the *number of levels* involved, while "level 0" refers to the 'name' of the level, which is the number of levels -1.

As shown in Figure 4, it is assumed that each class step can be preceded by a step executed by the environment (grey primitives), i.e. by some other class generating system interacting with the construction process. Thus, quite appropriately and realistically, a change in the environment (i.e. in some of its classes) may change the class elements, without an attendant change in the class generating system itself. Such concept of class admits the effects of the environment in a natural manner.

## Level 1 Structs

Suppose that an agent has already learned several level 0 classes, which together form the current **level 0 class setting**. Now, when representing an object, the agent has an access to a more refined form of object representation than a plain level 0 struct: it can now see if the struct is composed of several familiar level 0 class elements (Figure 5). This leads to the concept of **the next level (level 1) struct**, which includes *extra representational information* as compared to the underlying level 0 struct itself (in the form of the corresponding partition).

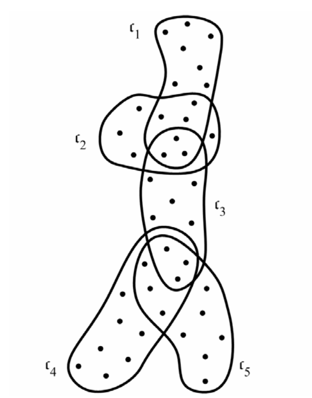

Figure 5: Simplified depiction of a level 1 struct in the contracted form: dots stand for primitives and solid lines delineate level 0 class elements $\mathfrak{c}_i$'s.

## Higher Level Classes

In the (recursive) **k-level** version of the **class representation**, for k ≥ 2, each corresponding step is specified by a set of **level (k - 1) constraints**. The level (k – 1) struct that is being attached at this step to the previously constructed part of the class element must now be composed only out of level (k – 2) admissible class elements[11] and satisfy one of the constraints specified for this step.

Similar to Figure 4, Figure 6 illustrates a (constructive) unit in such a generating process for a level 1 class element.

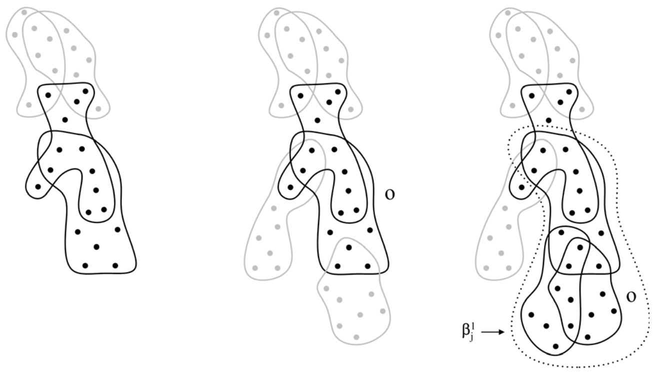


Figure 6: Illustration of a generic two-step unit in the construction of a level 1 class element. Dots stand for primitives, and solid lines delineate level 0 class elements, which now serve as basic constructive units. A step by the environment (bottom gray addition in the second struct) is followed by a step made by a level 1 class generating system itself (substruct $\beta_j^1$ of the third struct partly overlaying the second struct, where the superscript 1 refers to the level).

Moving one level up, for a level 2 class element, such element is an output of a level 2 (or three-levels) class generating system, where at each step the corresponding part of this level 2 class element is assembled out of several level 1 class elements and must satisfy one of the level 2 constraints specified for this step.

## Transition to the Next Representational Stage

To clarify the nature of a primitive, I outline very briefly the concept of the next stage of representation.

Transition to the next representational stage is associated with a representational compression, in which certain recurring (global) patterns of process interactions, called **transformations**, are compressed into new primitive transformations (for the next stage): for each of such global patterns, each of the interacting processes is compressed into a primal process and the segment in which the interaction between the processes occurs (the transform's 'body') is compressed into a next stage primitive event (Figure 7).

Note that a transformation is associated with a *disruption* (and the consequent restructuring) of the regular flow of several adjacent processes. In contrast, the case when several processes simply overlap, without being disrupted, is handled by the operation called *struct assembly* (not introduced here).

[11] Each of those must come from a class belonging to a (previously learned or given) set of level (k – 2) classes, comprising the current level (k – 2) class setting.

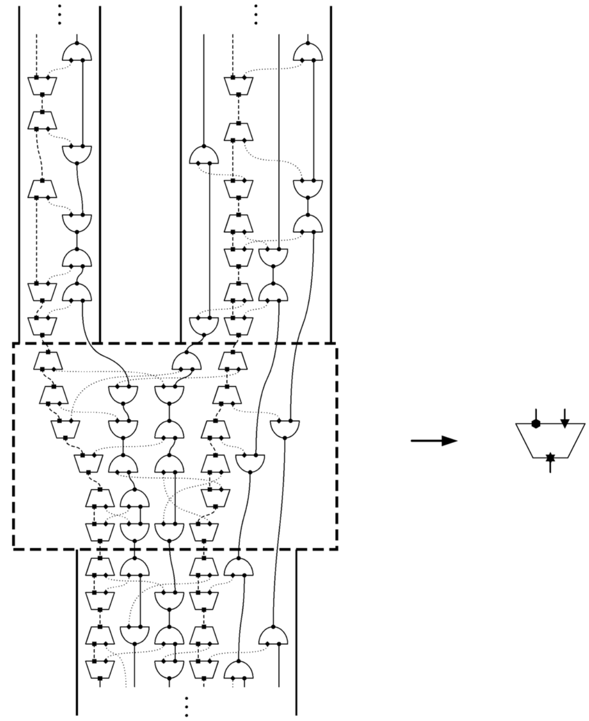

Figure 7: A transformation (left) and the corresponding next-stage primitive (right). An illustration of a transformation corresponding to a (hypothetical) formation of a lithium hydride molecule (terminal process) from hydrogen (left) and lithium (right) initial processes. The four primitives involved represent emission/absorption of a photon by electron (semi-circles) or nucleus (trapezoids). The body of the transform (heavy dashed line) depicts an imaginary restructuring of the two initial processes into the terminal one. On the right, the corresponding next-stage primitive (lithium hydride formation) is shown.

Thus, the primitives at the next representational stage include compressed transformations from the present stage and possibly some primitives from the present stage (that are simply lifted to the next one), resulting, for the fist time, in a *seamless integration of the stages* into the formalism.

## How Are We to Understand the Congruence of Syntax and Semantics in a Representational Formalism?

We are now ready to discuss in which sense the ETS exemplifies the representational formalism that possesses the above CSS property, i.e. the congruence of syntax and semantics. Indeed, as I suggested in the second section, we can see that the formal *structure* of the agent's representational formalism (syntax) is identical with formal structure of the agent's environment (semantics), since in ETS there is no *structural* distinction between the 'physical' (actual) object representation and the agent's representation.

This is because in the formalism, by its very design, *the principles* of object formation in nature, i.e. 'true' object representation, are exactly the same as those of object representation by an agent. Of course, what makes this structural identity possible is the chosen *event-based form of structural object representation*, associated with the formative object 'history'.

For example, consider a flower: its 'actual representation' is formed by the events in its full evolutionary/ developmental history, while its representation in an agent's 'mind' is formed by the perceptual events associated with the agent's (sensory) exploration of this flower. If the agent's sensory mapping is at all 'reasonable'—i.e. if for some fixed representational stage of the actual flower, each *sensory primitive event* captures some structural aspect of the struct representing the actual flower—then there must be a reasonable correlation between the two representations (starting from a particular stage of actual flower representation).

The identity of the two representational *structures* is a necessary but not sufficient condition for the CSS property: an arbitrary sensory mapping (an unlikely case) *can* botch the correspondence between the two representations.

## Conclusion

In this paper I wanted to draw attention to the serious challenge arising from the analysis of Searle's Chinese Room argument, namely, *the need for a representational formalism in which syntax and semantics are 'congruent'*. Moreover, I proposed to understand this 'congruence' as a structural identity of the *two complementary forms of object representation*: external, or complete, object representation (actual formative object 'representation', independent of any agent) and internal, or agent's, object representation.

In the event-based representational formalism outlined here—where the *objects are viewed and represented as structural* processes—the external object representation is formed by *all* events in the life of the object, while the internal representation is formed by the events comprising the agent's sensory interaction with the object. (This suggests, in particular, that it is an *event-based representational formalism* that allows one to remove the dualism between the syntax and semantics.)

Additionally, in view of its temporality, the proposed event-based object representation carries more information as compared to the numbers, strings, and graphs; in fact, it contains sufficient information to discern the *formative, or generative,* similarity of several objects that form the same category. In this light, it becomes clear why the generative program in cognitive science[12] could not be successfully carried out on the basis of the current representational formalisms: the *linear* sentence/string structure does not capture/represent its own formative history.

[12] The importance of this program Chomsky and a number of his followers, including Ray Jackendoff, e.g. Jackendoff (1994), have been, quite rightly, insisting on.

As far as neuroscience is concerned, the basic building block of ETS—the primitive event—has a structure resembling that of a neuron but suggesting a novel neuronal function: a particular neuron might be capturing a particular class of events, each transforming the 'regular' sensory flow of several adjacent processes.

The main point, however, is to take seriously the possibility that a representational formalism satisfying the syntax/semantics congruence property may indeed form the ultimate foundation for the development of cognitive science, since then we would be able to forget about this ubiquitous dichotomy—as well as many other 'basic' issues, including the "symbol grounding problem"—and proceed to study cognition within a unified scientific language.

## Acknowledgments

All figures are adopted from Goldfarb et al. (2007). I thank Muhammad Al-Digeil, Maria Goldfarb, Oleg Golubitsky, Dmitry Korkin, and Reuben Peter-Paul for help with the proofreading.

## References

Bringsjord, S., & Noel, R. (2002). Real robots and the missing thought-experiment in the Chinese room dialectic. In J. Preston & M. Bishop (Eds.), *Views into the Chinese room: New Essays on Searle and artificial intelligence*. Oxford, UK: Clarendon Press.

Goldfarb, L. (1990). On the foundations of intelligent processes I: An evolving model for pattern learning. *Pattern Recognition*, *23*, 595-616.

Goldfarb, L., & Nigam, S. (1994). The unified learning paradigm: A foundation for AI. In V. Honavar and L. Uhr (Eds.), *Artificial Intelligence and Neural Networks: Steps towards Principled Integration*. Boston, MA: Academic Press.

Goldfarb, L., Gay, D., Golubitsky, O., Korkin, D., Scrimger, I. (2007). *What is a structural representation? A proposal for an event-based representational formalism,* Sixth Variation (Tech. Rep. TR07-187). Fredericton, NB: University of New Brunswick, Faculty of Computer Science. http://www.cs.unb.ca/~goldfarb/ETSbook/ETS6.pdf

Harnad, S. (2002). Minds, machines, and Searle 2: What's right and wrong about the Chinese room Argument. In J. Preston & M. Bishop (Eds.), *Views into the Chinese room: New Essays on Searle and artificial intelligence*. Oxford, UK: Clarendon Press.

Jackendoff, R. (1994) *Patterns in the Mind: Language and Human Nature*. New York: Basic Books.

Preston, J., & Bishop, M. (Eds.) (2002). *Views into the Chinese room: New Essays on Searle and artificial intelligence*. Oxford, UK: Clarendon Press.

Rescher, N. (2008). Process Philosophy. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy*, forthcoming URL = <http://plato.stanford.edu/archives/spr2008/entries/process-philosophy/>.

Whitehead, A. N. (1978). *Process and reality: An essay in cosmology*, corrected edition. New York: The Free Press.