

# The reductionist blind spot

**Russ Abbott**

Department of Computer Science, California State University, Los Angeles, California  
[Russ.Abbott@GMail.com](mailto:Russ.Abbott@GMail.com)

**Abstract.** Can there be higher level laws of nature even though everything is reducible to the fundamental laws of physics? The computer science notion of level of abstraction explains how there can be. The key relationship between elements on different levels of abstraction is not the is-composed-of relationship but the implements relationship. I take a scientific realist position with respect to (material) levels of abstraction and their instantiation as (material) entities. They exist as objective elements of nature. Reducing them away to lower order phenomena produces a reductionist blind spot and is bad science.

**Key words:** emergence, entities, level of abstraction, reductionism

## 1. Introduction

When a male Emperor penguin stands for two frigid months balancing an egg on its feet to keep it from freezing, are we to understand that behavior in terms of quarks and other fundamental particles? It seems unreasonable, but that's the reductionist position. Here's how Albert Einstein [1] put it.

The painter, the poet, the speculative philosopher, and the natural scientist ... each in his own fashion, tries to make for himself .. a simplified and intelligible picture of the world.

What place does the theoretical physicist's picture of the world occupy among these? ... In regard to his subject matter ... the physicist ... must content himself with describing the most simple events which can be brought within the domain of our experience ... . But what can be the attraction of getting to know such a tiny section of nature thoroughly, while one leaves everything subtler and more complex shyly and timidly alone? Does the product of such a modest effort deserve to be called by the proud name of a theory of the universe?

In my belief the name is justified; for the general laws on which the structure of theoretical physics is based claim to be valid for any natural phenomenon whatsoever. *With them, it ought to be possible to arrive at ... the theory of every natural process, including life, by means of pure deduction. .... The supreme task of the physicist is to arrive at those elementary universal laws from which the cosmos can be built up by pure deduction.* [emphasis added]

The italicized portion expresses what Anderson [2] calls (and rejects) the *constructionist hypothesis*: the idea that one can start with physics and reconstruct the universe.

More recently Steven Weinberg [3] restated Einstein's position as follows.

Grand reductionism is ... the view that all of nature is the way it is ... because of simple universal laws, to which all other scientific laws may in some sense be reduced. ...

Every field of science operates by formulating and testing generalizations that are sometimes dignified by being called principles or laws. ... But there are no principles of chemistry that simply stand on their own, without needing to be explained reductively from the properties of electrons and atomic nuclei, and ... there are no principles of psychology that are free-standing, in the sense that they do not need ultimately to be understood through the study of the human brain, which in turn must ultimately be understood on the basis of physics and chemistry.

Not all physicists agree with Einstein and Weinberg. As Erwin Schrödinger [4] wrote,

[L]iving matter, while not eluding the 'laws of physics' ... is likely to involve 'other laws,' [which] will form just as integral a part of [its] science.

In arguing against the constructionist hypothesis Anderson [2] extended Schrödinger's thought.

[T]he ability to reduce everything to simple fundamental laws ... [does not imply] the ability to start from those laws and reconstruct the universe. ...

At each level of complexity entirely new properties appear. ... [O]ne may array the sciences roughly linearly in [a] hierarchy [in which] the elementary entities of [the science at level n+1] obey the laws of [the science at level n]: elementary particle physics, solid state (or many body) physics, chemistry, molecular biology, cell biology, ..., psychology, social sciences. But this hierarchy does not imply that science [n+1] is 'just applied [science n].' At each [level] entirely new laws, concepts, and generalization are necessary.

Notwithstanding their disagreements, all four physicists (and of course many others) agree that everything can be reduced to the fundamental laws of physics. Here's how Anderson put it.

[The] workings of all the animate and inanimate matter of which we have any detailed knowledge are ... controlled by the ... fundamental laws [of physics]. ... [W]e must all start with reductionism, which I fully accept.

Einstein and Weinberg argue that that's the end of the story. Starting with the laws of physics and with sufficiently powerful deductive machinery one should be able to reconstruct the universe. Schrödinger and Anderson disagree. They say that there's more to nature than the laws of physics—but they were unable to say what that might be.

Before going on, you may want to answer the question for yourself. Do you agree with Einstein and Weinberg or with Schrödinger and Anderson? Is there more than physics—and if so, what is it?

Note: I'd like this paragraph ("Before going on ...") to be set off by extra white space before and after or in some other way. Thanks.

The title and abstract of this paper give away my position. I agree with Schrödinger and Anderson. My position is that the computer science notion of *level of abstraction* explains how there can be higher level laws of nature—even though everything is reducible to the fundamental laws of physics. The basic idea is that a level of abstraction has both a specification and an implementation. The implementation is a reduction of the specification to lower level functionality. But the specification is independent of the implementation. So even though a level of abstraction depends on lower level phenomena for its realization it cannot be reduced to that implementation without losing something important, namely the properties that derive from its specification.

## 2. Levels of abstraction

A level of abstraction (Guttag [5]) is (a) a collection of types (which for the most part means categories) and (b) operations that may be applied to entities of those types. A standard example is the stack, which is defined by the following operations.

push(stack: s, element: e)	— Push an element e into a stack s and return the stack.
pop(stack: s)	— Pop the top element off the stack s and return the stack.
top(stack: s)	— Return (but don't pop) the top element of a stack s.

Although the intuitive descriptions are important for us as readers, all we have done so far is to declare a number of operations. How are their meanings defined? Axiomatically.

top(push(stack: s, element: e)) = e.	— After e is pushed onto a stack, its top element is e.
pop(push(stack: s, element: e)) = s.	— After pushing e onto s and then popping it off, s is as it was.

Together, these declarations and axioms define a stack as anything to which the operations can be applied while satisfying the axioms.

This is similar to how mathematics is axiomatized. Consider the non-negative integers as specified by Peano's axioms.<sup>1</sup>

1. *Zero* is a *number*.
2. If *A* is a *number*, the *successor* of *A* is a *number*.
3. *Zero* is not the *successor* of a *number*.
4. Two *numbers* of which the *successors* are equal are themselves equal.

---

<sup>1</sup> As given in Wolfram's MathWorld: <http://mathworld.wolfram.com/PeanosAxioms.html>.

5. (Induction axiom) If a set  $S$  of *numbers* contains *zero* and also the *successor* of every *number* in  $S$ , then every *number* is in  $S$ .

These axioms specify the terms *zero*, *number*, and *successor*. Here *number* is a type, *Zero* is an entity of that type, and *successor* is an operation on *numbers*. These terms stand on their own and mean (formally) no more or less than the definitions say they mean.

Notice that in neither of these definitions were the new terms defined in terms of pre-existing terms. Neither a *number* nor a *stack* is defined as a special kind of something else. Both Peano's axioms and the stack definition define terms by establishing relationships among them. The terms themselves, *stack* and a *number*, are defined *ab initio* and solely in terms of operations and relationships among those operations.

This is characteristic of levels of abstraction. When specifying a level of abstraction the types, objects, operations, and relationships at that level stand on their own. They are not defined in terms of lower level types, objects, operations, and relationships.

See the sidebar on how levels of abstraction function in different disciplines.

### 3. *Unsolvability and the Game of Life*

The Game of Life<sup>2</sup> is a 2-dimensional cellular automaton in which cells are either alive (on) or dead (off). Cells turn on or off synchronously in discrete time steps according to rules that specify cell behavior as a function of their eight neighbors.

- Any cell with exactly three live neighbors will stay alive or become alive.
- Any live cell with exactly two live neighbors will stay alive.
- All other cells die.

The preceding rules are to the Game-of-Life world as the fundamental laws of physics are to ours. They determine everything that happens on a Game-of-Life grid.

Certain on-off cell configurations create patterns—or really sequences of patterns. The glider is the best known. When a glider is entered onto an empty grid and the rules applied, a series of patterns propagates across the grid. Since nothing actually moves in the Game of Life—the concept of motion doesn't even exist—how should we understand this?

---

<sup>2</sup> An accessible popular discussion of the Game of Life is available in Poundstone [6].

Gliders exist on a different level of abstraction from that of the Game of Life. At the Game-of-Life level there is nothing but grid cells—in fixed positions. But at the glider level not only do gliders move, one can even write equations for the number of time steps it will take a glider to move from one location to another. What is the status of such glider velocity equations?

Before answering that question, recall that it's possible to implement Turing machines by arranging gliders and other Game-of-Life patterns. Just as gliders are subject to the laws of glider equations, Turing machines too are subject to their own laws—in particular, computability theory.

Game-of-Life gliders and Turing machines exemplify the situation described by Schrödinger. They are phenomena that appear on a Game-of-Life grid but are governed by laws that apply on a different and independent level of abstraction. While not eluding the Game-of-Life rules, autonomous new laws apply to them. These additional laws are not expressible in Game-of-Life terms. There is no such thing as a glider or a Turing machine at the Game-of-Life level. The Game of Life is nothing but a grid of cells along with rules that determine when cells go on and off. In other words, Game-of-Life gliders and Game-of-Life Turing machines (a) are governed by laws that are independent of the Game of Life rules while at the same time they (b) are *completely determined* by the Game of Life rules.

#### ***4. Evolution is also a property of a level of abstraction***

Evolution offers another example of how levels of abstraction give rise to new laws. Evolution is an abstract process that can be described as follows.

Evolution occurs in the context of a *population* of *entities*. The entities exist in an *environment* within which they may *survive* and *reproduce*. The entities have *properties* that affect how they *interact* with their environment. Those interactions help determine whether the entities will survive and reproduce. When an entity *reproduces*, it *produces offspring* which *inherit* its properties, possibly along with some *random variations*, which may *result in new properties*. In some cases, pairs of entities *reproduce jointly*, in which case the offspring *inherit* some *combination* of their parent's properties—perhaps also with *random variations*.

The more likely an entity is to survive and reproduce, the more likely it is that the properties that enabled it to survive and reproduce will be passed on to its offspring. If certain properties—or random variations of those properties, or the random creation of new properties—enable their possessors to survive and reproduce more effectively, those properties will propagate.

We call the generation and propagation of successful properties *evolution*. By helping to determine which entities are more likely to survive and reproduce, the

environment selects the properties to be propagated—hence evolution by environmental (i.e., natural) selection.

The preceding description introduced a number of terms (in italics). As in the case of stacks and Peano numbers, the new terms are defined *ab initio* at the evolution level of abstraction. The independent usefulness of evolution as a level of abstraction is illustrated by evolutionary computation, which uses the abstract evolutionary mechanism to solve difficult optimization problems. It does so in a way that has nothing to do with biology or natural environments.

## 5. *The reductionist blind spot*

Physics recognizes four fundamental forces. Evolution is not one of them. Similarly there is no computational functionality in a Game-of-Life universe. In other words, both evolution and Turing machine computation appear as phenomena within frameworks that are blind to their existence. Nevertheless, both evolution and Turing machine computation can be completely explained in terms of phenomena that operate as primitives within those frameworks. Given that, do we really need concepts such as evolution and Turing machine computation?

In some sense we don't. Echoing Kim [7], Schouten and de Jong [8] put it this way.

If a higher level explanation can be related to physical processes, it becomes redundant since the explanatory work can be done by physics.

In this sense both evolution and computations done by Game-of-Life Turing machines are redundant. After all, Game-of-Life Turing machines as such don't *do* anything. It is only the Game-of-Life rules that make cells go on and off. Reductionism has not been overthrown. One could trace the sequence of Game-of-Life rule applications that transform an initial Game-of-Life configuration (that could be described as a Turing machine with input  $x$ ) into a final configuration (that could be described as a Turing machine with output  $y$ ). One could do this with no mention of Turing machines.

Similarly one could presumably—albeit with great difficulty—trace the sequence of chemical and physical reactions and interactions that produce a particular chemical configuration (that could be described as the DNA that enables its possessor to thrive in its environment). One could do this with no mention of genes, codons, proteins, or other evolutionary or biological terms.

One can always reduce away macro-level terminology and associated physical phenomena and replace them with the underlying micro-level terminology and associated physical phenomena. It is still the elementary mechanisms—and nothing but those mechanisms—that turn the causal crank. So why not reduce away higher levels of abstraction?

Reducing away a level of abstraction produces a reductionist blind spot. Computations performed by Game-of-Life Turing machines cannot be described as computations when one is limited to the vocabulary of the Game-of-Life. Nor can one explain why the Game of Life halting problem is unsolvable. These concepts exist only at the Turing machine level of abstraction. Similarly, biological evolution cannot be explicated at the level of physics and chemistry. The evolutionary process exists only at the evolution level of abstraction. It is only entities at that level of abstraction that evolve.

Furthermore, reducing away a level of abstraction throws away elements of nature that have objective existence. At each level of abstraction there are entities (see Section 10)—such as Turing machines and biological organisms—that instantiate types at that level. These entities are simultaneously causally reducible and ontologically real—a formulation coined by Searle [9] in another context. Entities on a level of abstraction that are implemented by a lower level of abstraction are causally reducible because the implementation provides the forces and mechanisms that drive them. But such entities are ontologically real because (a) their specifications, which are independent of their implementations, characterize what they do and how they behave and (b) they are objectively observable, i.e., observable independently of human conceptualization as a result (i) of their reduced entropy and (ii) of their mass distinctions. Again, see Section 10 for additional discussion of entities.

The goal of science is to understand nature. Reducing away levels of abstraction discards both real scientific explanations—such as the evolutionary mechanism—and objectively real entities—such as biological organisms. Denying the existence of biological organisms as entities requires that one also throw away biological taxonomic categories such as species, or phyla, or even kingdoms. What are such categories after all if there are no such things as biological entities for them to collect? But do we really want to dismiss the grand taxonomy of life—with a place for all life forms from *E. coli* to elephants—whose structure and history biology has been so successful in describing? What would be left of biology? Not much. Reducing away levels of abstraction and the entities associated with them is simply bad science.

Reducing away levels of abstraction is bad science from an information theoretic perspective as well. Chaitin [10] points out that Leibniz anticipated algorithmic information theory when he characterized science as developing the simplest hypothesis (in the algorithmic information theory sense) for the richest phenomena. Throwing away a level of abstraction typically increases the algorithmic complexity of a description of some phenomenon.<sup>3</sup>

---

<sup>3</sup> Dennett [11] makes a similar observation. See the Appendix for an extended discussion of that article.

## 6. *Constructionism and the principle of ontological emergence*

Game-of-Life Turing machines and biological evolution illustrate Schrödinger's insight that although higher level phenomena don't elude the laws of physics they are governed by new laws. Because the higher level laws are not derived from the laws governing the implementing level, knowledge of the lower level laws does not enable one to generate a specification and implementation of the higher level. That is, one would not expect to be able to deduce computability theory from knowledge of the Game-of-Life rules, and one would not expect to be able to deduce biological evolution from knowledge of fundamental physics. As Anderson argued—and contrary to Einstein—constructionism fails. No matter how much deductive power one has available, one should not expect to start with the fundamental laws of physics and reconstruct all of nature.

In some ways the preceding statement is a bit of an exaggeration. Computability theory, after all, can be derived from first principles. Since the rules of the Game of Life are not incompatible with the theory of computability, throwing them in as extra premises doesn't prevent that derivation.

The point is that higher level abstractions are typically creative additions to lower levels. The notion that one could start with lower level elements and deduce higher level elements is similar to the notion that one could start with a mountain of granite and deduce the faces of Washington, Jefferson, Lincoln, and Roosevelt. The granite can be carved and molded into those faces. But given the intuitive interpretation of *deduce* it makes little sense to say that one could start with the granite and deduce the faces. The idea of carving those faces into the granite was a creative leap, not what would normally be considered a deduction.

Even with this in mind, though, constructionism can be said to succeed. It has taken billions of years, but nature *has* implemented biological organisms. And the faces of Washington, Jefferson, Lincoln, and Roosevelt, *are* on Mt. Rushmore. Nature accomplished this trick starting from quantum mechanics. So if one considers nature as a mechanism for generating and implementing new levels of abstraction, then nature embodies constructionism.

Nature does its work as a random enumerator of possibilities—and not in the deductive/explanatory sense suggested by Einstein and Weinberg. Nature is both creative in the sense of Dennett [12] and constructive. Normally one doesn't refer to nature's processes as deductive. But just as software theorem provers work by searching the space of possible proofs until they find one that works, nature too proceeds by search, retaining levels of abstraction that work and discarding those that don't. If software theorem provers are deductive, then so is nature.

Nature is continually generating new levels of abstractions. Which persist? It depends on the environment at the time. Molecules persist only in environments with low enough temperatures; biological organisms persist only in environments

that provide nourishment; and hurricanes (the only non-biological and non-social dynamic entity of which I'm aware) persist only in environments with a supply of warm water. This can be summarized as the principle of ontological emergence.

*Extant levels of abstraction are those whose implementations have materialized and whose environments enable their persistence.*<sup>4</sup>

It's important to realize, though, that in generating new levels of abstraction nature does not build strictly layered hierarchies. New entity types may interact with any existing entity type. The levels are not partitioned into disjoint layers that interact only hierarchically. This is nicely illustrated by the fact that the gecko, a very macro organism, makes direct use (Kellar [14]) of the quantum level van der Waals force to cling to vertical surfaces.

## ***7. Constraints, predictions, and downward entailment***

Higher level laws generally have lower level implications. Because the halting problem is unsolvable, for example, it is unsolvable whether an arbitrary Game-of-Life configuration will ever reach a stable state. And because the Game of Life can implement a Turing machine, the Game of Life can compute any computable function. In other words, computability theory, a law that applies to a Game-of-Life Turing machine, has consequences for the Game of Life itself. Similarly, velocity equations for Game-of-Life gliders can be used to predict when a glider will "turn on" a particular cell.

A similar phenomenon illustrates how the abstract theory of evolution predicts DNA—or something like it. When Darwin and Wallace described the evolution level of abstraction, they knew nothing about DNA. But their model required some mechanism for recording and transmitting properties. In other words, their model made a prediction that any implementation of the evolution level of abstraction must provide a mechanism for transmitting properties from parents to offspring. Because biology implements the evolutionary level of abstraction one

---

<sup>4</sup> I treat levels of abstraction—and their instantiations as entities—as real elements of nature. This contrasts with the position taken by Floridi [13], which treats levels of abstraction as epistemological. The focus of Floridi's work is to understand observable data in terms of typed variables, which in turn are organized as levels of abstraction. As Floridi writes in his conclusion,

I have shown how ... analysis ... may be conducted at different levels of epistemological abstraction without assuming any corresponding ontological levelism. Nature does not know about [Levels of Abstraction] ...

My position is that not only does nature know about levels of abstraction, they are fundamental to how nature builds the richness we see around us.

can conclude that biological organisms must have such a means to transmit properties. We now know that DNA is that mechanism. Prediction confirmed.

When autonomous higher level laws apparently affect lower level phenomena the result has been called (Andersen [15]) downward causation. But downward causation doesn't make scientific sense. It is always the lower level phenomena that determine the higher level. The Game-of-Life rules, not glider equations, are the only things that determine when and whether cells go on and off.

But if causation is always upwards, how can computability theory and glider equations let us draw conclusions about Game-of-Life cells? How can evolution let us draw conclusion about biological organisms? In [16] I call this *downward entailment*. Autonomous laws that apply at a higher level of abstractions can have implications for elements at a lower level as long as the lower level is implementing the higher level.

When frozen into ice cubes, for example, H<sub>2</sub>O molecules form a rigid lattice and are constrained to travel together whenever the ice cube that they implement is moved about. This is only common sense. As long as molecules of H<sub>2</sub>O are implementing a solid, they are constrained by laws that govern solids. Once the ice cube melts and the H<sub>2</sub>O molecules are no longer implementing a solid, they are no longer bound by the laws of solids. This clarifies the somewhat mystical-sounding position taken by Sperry [17] when discussing how it is that the atoms and molecules that make up a wheel move in such a coordinated way.

The fate of the entire population of atoms, molecules, and other components [that constitute a wheel rolling downhill] are determined very largely by the holistic properties of the whole wheel as a unit.

In these examples, constraints play a primary role. They may be seen to be operating in two directions. First, the lower level system is constrained so that it implements some higher level abstractions. The Game of Life is constrained to behave like a Turing machine; water molecules are constrained to behave as a solid; granite is constrained to form the features of four American presidents.

Once those constraints are in place, the properties of the higher level objects constrain the implementing components. This second sort of constraint may be misleading if it suggests downward causation. There is no downward causation. But the properties and behaviors of the higher level object necessitate properties or behaviors of the lower level elements that implement them. As long as the lower level continues to implement the upper level the lower level is necessarily constrained by whatever constraints apply at the upper level.

### ***8. The fundamental relationship between levels of abstraction***

Putnam [18] makes a similar argument. He asks how one should explain why a square peg won't fit into a round hole whose diameter is the same length as the peg's side. Should the explanation be based on quantum physics or on geometry? Putnam's answer is that the explanation should be based on geometry.

An explanation at the level of quantum physics explains only the one particular peg-and-hole pair under consideration whereas one based on geometry explains all peg-and-hole pairs of incompatible dimensions. Putnam argues in particular that the quantum-level explanation must consider the particular elementary particles (and hence the materials) of which the peg and hole are made. But the particular particles and materials are not (or should not be) relevant—as long as they implement non-deformable materials; only the relative dimensions of the peg and hole matter. Thus the geometrical explanation is superior.

When considering the peg-and-hole question, one of the fundamental issues concerns the language and concepts one should allow oneself to use. At the quantum level, there is no such thing as a peg. So how can one even begin to approach the question? A peg can only be laboriously constructed by describing how it is constructed from elementary particles. But if one then makes an argument based on the geometry of the constructed peg is one not still using the argument from geometry rather than the argument from quantum physics? The argument from quantum physics would have to focus on the individual particles in the peg and the hole. But in doing that, one would have lost track of the peg and hole as geometrical entities—which are the subject matter of the original question.

The fundamental relationship between levels of abstraction is the implementation relation: one level implements another. An argument that describes how a peg and hole may be implemented from quantum phenomena and then claims based on the geometry of the resulting peg and hole that one cannot be inserted into the other is really making an argument at the geometric level. The only role that the quantum level plays is to show that it is possible to implement pegs and holes using quantum phenomena.

On the other hand, if one does not construct a peg and a hole from quantum phenomena but simply shows that a particular configuration of elementary particles (that we would describe as a peg) and another configuration of elementary particles (that we would describe as a hole) cannot be manipulated so that they would fit the description that we would call having the peg inside the hole, then one must make that argument for every configuration of elementary particles that one wishes to cover. Even then, it isn't clear how one could claim that one has said anything about pegs and holes in general or that one could even define the terms peg and hole.

One might approach the problem from a different direction. Since at the quantum level one can make use of spatial language, one can define peg and hole shapes of the appropriate dimensions. One could then argue that if these shapes are presumed not to be inter-penetrable, then the peg shape could not be positioned within the hole shape. One would then describe how such shapes could be filled with quantum material so that they become non-inter-penetrable, i.e., solids. But in doing so, isn't one again showing how one could use the quantum level to implement pegs and holes and then making a geometrical argument?

It seems to me that any argument showing that a peg and hole of incompatible sizes cannot fit one within the other must be made at the geometrical level and that when one starts at the quantum level, one finds oneself describing how to implement the level of geometrical solids and then making the argument at the geometrical level. Perhaps the problem is that one simply cannot talk about pegs and holes in any language other than at the level of geometrical abstractions.

### ***9. Levels of abstraction and multiple realizability***

The peg-and-hole and similar examples are frequently used to argue the functionalist position that multiply realizable properties are not reducible: if there are multiple realizations, to which one is the higher level property reducible? I believe that this argument misses the point. A level of abstraction, like a Turing machine, exists at the abstraction level because it is independently specifiable—not because it is multiply realizable, e.g., as a Game-of-Life and as a Turing machine implemented on some other platform. The abstraction exists as an abstraction whether or not it is realized. Similarly a level of abstraction with only one implementation is just as real an abstraction as one with multiple realizations.

How are abstractions related to what actually exists in the world? The perspective I favor turns the question of realizability around. In exploring what actually exists the question is not whether any particular abstraction is multiply realizable. The question becomes what new levels of abstraction can one implement given the currently existing levels of abstraction? Does it really matter, for example, whether eyes or wings evolved once or multiple times? What really matters is that each time they evolved they enabled its possessors to see or fly. The fact (if it is a fact) that vision and flight are more or less the same in each case is not important. What is important is that a vision/flight capability was created, whether that happened once or many times. The ontological status of higher levels of abstraction should not be dependent on how often those abstractions have been realized.

The preceding is not intended to deny that levels of abstraction can be implemented in multiple ways. One of the fundamental tenants in my own field of computer science is the importance of distinguishing between a specification and an implementation. It is the specification that determines how something can be ex-

pected to behave. It doesn't matter how that specification is implemented as long as the implementation realizes the specification. There can be multiple ways to implement a specification. So it is certainly possible for a level of abstraction, i.e., a specification, to have multiple implementations. But it is not the multiple implementations that make the specification independent of the implementation. It is the specification itself, the fact that it can be expressed without relying on the implementation as part of the description that makes it independent.

Specifications typically occur in the context of man-made artifacts. Nature neither writes nor implements specifications. But nature does provide environments that are more manageable when entities have certain features and capabilities. In that sense one can think of nature as providing specifications: the environment that must be navigated. Since vision and flight are useful capabilities, the ability to focus and to extract information from light and the ability to suspend oneself and propel oneself through the air can be understood as specifications for vision and flight capabilities. Those "specifications" may each have been implemented once—with a number of variations, or they may each have been implemented multiple times—resulting in a number of similar capabilities in different organisms. Once or many times doesn't matter; it's the ability to see or to fly that matters.

## 10. Entities

Although it hasn't been raised explicitly, central to this discussion is the issue of entities. Are there higher level entities? What is the ontological status of instances of levels of abstraction? This section discusses these questions.

In this article I'm considering only material entities. Other entity-like elements such as time instants and durations, geometric regions, numbers, etc. are beyond the scope of this discussion. Provisionally I'll define an entity as a persistent *pattern*. Since a pattern implies increased organization, an entity is an area of reduced entropy.

I wish I could think of a better word than "area." But I have not been able to come up with a category of which entity is a subcategory. Perhaps that means that entity is itself a level of abstraction. This is consistent with the software world. In many object-oriented programming languages, the most general type is an undifferentiated "object." Object is a primitive of the language; it is not defined in terms of anything else. So perhaps we should take entities as primitive and simply note that we identify them because they persist and have reduced entropy.

Reduced entropy implies energy. So entities must be related to energy. Table 1 presents a categorization of entities according to two characteristics: energy and whether they are naturally occurring or man-made.

Table 1. Categories of entities

		<i>Naturally occurring</i>	<i>Human designed</i>
<i>Energy Status</i>	<b>Static.</b> At an energy equilibrium; in an “energy well.” Supervenience is useful.	Atoms, molecules, solar systems, ...	Tables, boats, houses, cars, ships, ...
	<b>Dynamic.</b> Must import energy (and usually other resources) to persist. Supervenience is not useful.	Hurricanes(!), biological organisms, biological groups, ...	Social groups such as governments, corporations, clubs, the ship of Theseus(!), ...
	<b>Subsidized.</b> Energy is not relevant since it is provided “for free” within a “laboratory” which has built-in support for entities.	Ideas, concepts, “memes,” ... The elements of a conceptual system. (This paper is not about consciousness. This category just fits here.)	The “first class” values—such as objects, classes, class instances, etc.—within a computational system.

**Static entities.** These are entities that exist in an energy well. Examples include atoms (made from elementary particles), molecules (made from atoms), solids (made from atoms and molecules), etc. As an instance of a level of abstraction an entity is the product of constraints. In this case, the constraints are the fundamental forces that hold components together. Phase transitions typically mark the imposition or removal of constraints of this sort. The forces that create static entities produce energy wells; energy is required to break them apart. Consequently a static entity comprises less mass as an entity than its components taken separately. Since the components of a static entity tend to remain identifiable as part of the entity, supervenience tends to be useful. Static entities supervene over their components. Naturally occurring static entities are those familiar to us from physics, chemistry, and the other “hard” sciences. Most human designed objects are also static entities.

**Dynamic entities.** These are entities in which components are tied together by procedural processes. Examples include biological organisms (naturally occurring) and social groups (man-made). A biological organism persists as long as its components interact in just the right ways. Similarly, the processes of a social group, i.e., the ways in which the group members behave and interact, cause the group to persist as a group. To take a very simple example a social club—a bridge club, a

bowling club, etc.—is held together by the fact that the members adhere to (formal or informal) agreements about how they will behave.

Dynamic entities have the interesting property that their components may change while the entity itself persists. People may join and leave a club even though the club persists. One is a member of the club as long as one behaves according to the processes that define the club. Those old enough to remember *Guys and Dolls* may recall “the oldest established permanent floating crap game in New York.” The same sort of analysis applies to animal groups like herds, colonies, etc. Similarly, biological organisms gain and lose molecules while they persist as organisms. This feature makes dynamic entities less amenable to analysis by supervenience. A dynamic entity typically does not supervene over the collection of components that make it up at any one time.

Wilson [19] makes the point that virtually everything in the social and biological realm is both a group and an entity. That echoes from an evolutionary perspective the point made in this paper, that entities at a level of abstraction must be understood in terms of their behavior at that level even though they also understood as being implemented by elements from lower levels.

In another contrast to static entities (which require energy to tear them apart) dynamic entities require energy to keep themselves together. They cohere—the dynamic entity persists—only so long as their components behave according to the rules that define how they should behave. Such behavior requires energy. Consequently, dynamic entities must import energy from their environments to persist. Because dynamic entities involve components in action, they comprise more mass (the components along with their energy of action) than their components separately.

**Symbolic entities.** A symbolic framework provides the means to create new abstractions. Entities created within such a framework are (appropriately) called symbolic entities. Examples are the entities created within computational frameworks such as the Game of Life and computer programming languages and environments. In symbolic frameworks, mechanisms exist to support the creation of new abstractions. No special energy is required as long as the framework itself continues to exist. Although the framework itself must have energy supplied to it, the individual entities are not strongly tied to energy. Presumably a similar mechanism (the symbolic framework that operates within our consciousness) enables us to conceptualize symbolic entities.

Static and dynamic entities are distinguishable by mass: static entities have less mass and dynamic entities have more mass than the mass of their components taken separately. All three classes of entities are distinguished by their entropy. They all have less entropy than their surroundings—and their components are more highly correlated with each other (a) than with outside elements and (b) than outside elements are with each other. Because of these objectively observable properties, entities are part of nature’s ontology, i.e., they are objectively real.

Furthermore, even though most if not all of these entities can be reduced to their components—one can describe in detail how their components fit together to produce the entity—they interact with their environment, including other entities, as entities. Nations go to war with each other; biological organisms breed; ships float and carry passengers and freight; etc. The descriptions of how entities interact as entities define their levels of abstraction. To reduce away those interactions is to deny the reality of the interacting entities.

## 11. Summary

The need to understand and describe complex systems led computer scientists to develop concepts that clarify issues beyond computer science. In particular, the notion of the level of abstraction and its implementation by pre-existing levels of abstraction explains how higher level laws of nature help govern a reductionist universe.

## Acknowledgements

I thank Debora Shuger for many stimulating conversations and the anonymous reviewer for helpful comments and suggestions.

## References

All Internet accesses are as of April 3, 2008.

1. Einstein, Albert, "Principles of Research," 1918 address to the Physical Society, Berlin, reprinted in *Ideas and Opinions*, Crown, 1954.  
[http://www.cs.ucla.edu/~slu/on\\_research/einstein\\_essay2.html](http://www.cs.ucla.edu/~slu/on_research/einstein_essay2.html).
2. Anderson, Philip W., "More Is Different," *Science*, 1972, (177, 4047), 4 Aug. 393-396.
3. Weinberg, Steven, "Reductionism Redux," *The New York Review of Books*, October 5, 1995. Reprinted in Weinberg, Steven., *Facing Up*, Harvard University Press, 2001.  
[http://www.idt.mdh.se/kurser/ct3340/ht02/Reductionism\\_Redux.pdf](http://www.idt.mdh.se/kurser/ct3340/ht02/Reductionism_Redux.pdf).
4. Schrödinger, Erwin, *What is Life?*, Cambridge University Press, 1944.  
<http://home.att.net/~p.caimi/Life.doc>.
5. Guttag, John, "Abstract data types and the development of data structures," *Communications of the ACM*, 1977, (20, 6), June 396-404.  
<http://rockfish.cs.unc.edu/204/guttagADT77.pdf>.
6. William Poundstone, *The Recursive Universe*, Oxford University Press, 1987.
7. Kim, Jaegwon, "Epiphenomenal and Supervenient Causation," *Midwest Studies in Philosophy*, 1984, 9, 257-70.

**Note.** A sidebar and an Appendix follow the references.

8. Schouten, Maurice and Huib Looren de Jong, *The Matter of the Mind*, Wiley-Blackwell, 2007.
9. Searle, John, *Mind: a brief introduction*, Oxford University Press, 2004.
10. Chaitin, Gregory, "Leibniz, Information, Math and Physics," *Wissen und Glauben / Knowledge and Belief. Akten des 26. Internationalen Wittgenstein-Symposiums 2003*, Herausgegeben von Löffler, Winfried / Weingartner, Paul, ÖBV & HPT, Wien, 2004, 277-286. <http://www.umcs.maine.edu/~chaitin/kirchberg.html>.
11. Dennett, Daniel C., "Real Patterns," *The Journal of Philosophy*, 1991, (88, 1), 27-51.
12. Dennett, Daniel C., *Darwin's Dangerous Idea*, Simon & Schuster 1995.
13. Floridi, Luciano, "The Method of Levels of Abstraction," *Minds and Machines*, Springer, 2008, (18.3), 303-329.
14. Kellar Autumn; Metin Sitti; Yiching A. Liang; Anne M. Peattie; Wendy R. Hansen; Simon Sponberg; Thomas W. Kenny; Ronald Fearing; Jacob N. Israelachvili; Robert J. Full. "Evidence for van der Waals adhesion in gecko setae." *Proceedings of the National Academy of Sciences of the USA*, 2002, 99, 12252-12256.
15. Andersen, Peter Bøgh, Claus Emmeche, Niels Ole Finnemann and Peder Voetmann Christiansen, eds.: *Downward Causation. Minds, Bodies and Matter*. Århus: Århus University Press, 2000.
16. Abbott, Russ, "Emergence explained," *Complexity*, Sep/Oct, 2006, (12, 1) 13-26. Preprint: [http://cs.calstatela.edu/wiki/images/9/95/Emergence\\_Explained-Abstractions.pdf](http://cs.calstatela.edu/wiki/images/9/95/Emergence_Explained-Abstractions.pdf)
17. Sperry, Roger, "An Objective Approach to Subjective Experience," *Psychological Review*, 77, 585-590, 1970.
18. Putnam, Hilary, "Philosophy and our Mental Life." In *Mind, Language, and Reality*. Cambridge University Press, 1975, 291-303.
19. Wilson, David Sloan, *Evolution for Everyone*, Delacorte Press, 2007.
20. Abbott, Russ, "Bits don't have error bars," *Selected papers from the 2007 Workshop on Philosophy and Engineering*, Springer Verlag, to appear.

### **(Sidebar) Mathematics, science, and engineering (including computer science)**

The notion of the level of abstraction clarifies some of the similarities and differences among mathematics, science, engineering, and computer science.

**Mathematics** is the study of the entities and operations defined on various levels of abstraction—whether or not those levels of abstraction are implemented. Mathematicians devise formal (or at least "rigorous") specifications of levels of abstraction. They then study the consequences of those specifications—which in the case of Peano's axioms is number theory.

**Science** is (a) the characterization of observed natural phenomena as levels of abstraction, i.e., the framing of observed phenomena as patterns, followed by (b) a determination of how those levels of abstraction are implemented by lower level mechanisms.

**Engineering** (including computer science) is the imagination and implementation of new levels of abstraction. The levels of abstraction that engineers and computer scientists implement are almost always defined informally—most real-world systems are too complex to specify formally. They are often characterized in terms of what are called *requirements*, natural language descriptions of required functional and performance properties. Engineers and computer scientists implement systems that meet requirements.

Whereas engineers and computer scientists imagine and implement new levels of abstraction, scientists identify existing levels of abstraction and discover the mechanisms nature uses to implement them. In other words, science is the reverse engineering of nature.

Why did computer science rather than engineering develop the notion of level of abstraction? In [21] I discuss how computer scientists start from a well defined base level of abstraction—the bit and the logical operations defined on it—and build new levels of abstraction upwards from that base. Engineers work with physical objects implemented at multiple and arbitrary levels of abstraction. Since there is no engineering base level of abstraction, engineers construct mathematical models that approximate nature as far down as necessary to ensure that the systems they build have reliable physical foundations. Engineers are often preoccupied with approximating downward. Given the (paradoxically solid) foundation of the bit, computer scientists have more freedom to imagine upward.

## Appendix. Dennett's "Real Patterns"

In "Real Patterns" Dennett (1991) uses the fact that a Turing Machine may be implemented in terms of Game-of-Life patterns to argue for his *The Intentional Stance* (1987) position regarding beliefs—which he calls mild realism.

It has been suggested that "Real Patterns" has a significant overlap with this paper. I disagree. But to explore that issue, this appendix examines "Real Patterns" in some depth. My primary goal is to describe (in Dennett's own words as much as possible) the primary points made in "Real Patterns."

The fundamental issue discussed in "Real Patterns" is the status of beliefs. Much of the paper draws connections among beliefs, patterns, and predictions. Here's an extract which is represented as the paper's core content. It appears (as of July 1, 2008) on the Tufts Cognitive Study website: <http://ase.tufts.edu/cogstud/papers/realpatt.htm>.

Are there really beliefs? Or are we learning (from neuroscience and psychology, presumably) that, strictly speaking, beliefs are figments of our imagination, items in a superseded ontology? Philosophers generally regard such ontological questions as admitting just two possible answers: either beliefs exist or they don't. There is no such state as quasi-existence; there are no stable doctrines of semi-realism. Beliefs must either be vindicated along with the viruses or banished along with the banshees. A bracing conviction prevails, then, to the effect that when it comes to beliefs (and other mental items) one must be either a realist or an eliminative materialist.

Dennett suggests that one way to evaluate a belief is by looking at predictions they allow one to make. He writes (p. 30) that "the success of any prediction depends on there being some order or pattern in the world to exploit. What is the pattern a pattern *of*?"

Thus, Dennett acknowledges—unsurprisingly—that there are regularities in the world, which he tends to call patterns. Dennett does not seem to be asking how those regularities come about or what they consist of. He seems more interested in the relationship between such regularities and how we think about them. Dennett continues (p. 30).

Some have thought, with Fodor, that the pattern of belief must in the end be a pattern of structures in the brain, formulae written in the language of thought. Where else could it be? Gibsonians might say the pattern is "in the light"—and Quinians (such as Donald Davidson and I) could almost agree: the pattern is discernible in agents' (observable) behavior when we subject it to "radical interpretation" (Davidson) "from the intentional stance" (Dennett).

When are the elements of a pattern real and not merely apparent? Answering this question will help us resolve the misconceptions that have led to the proliferation of "ontological positions" about beliefs, the different grades or kinds of realism. I shall concentrate on five salient exemplars arrayed in the space of possibilities: Fodor's industrial strength Realism . . . ; Davidson's regular strength realism; my mild realism; Richard Rorty's milder-than-mild irrealism, according to which the pattern is only in the eyes of the beholders, and Paul Churchland's eliminative materialism, which denies the reality of beliefs altogether."

Earlier (p. 29) he writes,

I have claimed that beliefs are best considered to be abstract objects rather like centers of gravity. ... My aim [in this paper] is not so much to prove that my intermediate doctrine about the reality of psychological states is right, but just that it is quite possibly right ... .”

His concern, he says (p. 30), is

not in differences of opinion about the ultimate metaphysical status of physical things or abstract things (e.g., electrons or centers of gravity), but in differences of opinion about whether beliefs and other mental states are, shall we say, *as real as* electrons or centers of gravity. I want to show that mild realism is the doctrine that makes the most sense when what we are talking about are real patterns, such as the real patterns discernible from the intentional stance.

Although earlier Dennett acknowledged that there are regularities (patterns) in nature, it is not clear from the final sentence in the preceding whether Dennett is now claiming that at least some of these patterns become apparent only when one takes the intentional stance.

In contrast to what seems like philosophical infighting, my concern is not with the ontological status of beliefs but with what I claim are real features of nature—whether anyone has beliefs about them or not.

Dennett cites an article by Chaitin that discusses Chaitin’s measure of randomness and says that a pattern is real “if there is a description of the data that is more efficient than the bit map,” i.e., more concise than a literal replication of the primitive elements of which the pattern is composed.

Yet this discussion about patterns and their efficient representation seems to be diluted by Dennett’s more general acknowledgement (p. 36) that science is “widely acknowledged as the final arbiter of ontology. Science is supposed to carve nature at the joints—at the *real* joints, of course.”

If that is the case, then to determine what is real, ask a scientist—or at least an expert in the field—who presumably has a more efficient (or more insightful?) way of describing data than an exhaustive enumeration.

And that is more or less the position that Dennett takes. Dennett then refers (p. 41) to the fact that a Turing machine can be built using Game-of-Life patterns.

Since the universal Turing machine can compute any computable function, it can play chess—simply by mimicking the program of any chess-playing computer you like. ... Looking at the configuration of dots that accomplishes this marvel would almost certainly be unilluminating to anyone who had no clue that a configuration with such powers could exist. But from the perspective of one who had the hypothesis that this huge array of black dots was a chess-playing computer, enormously efficient ways of predicting the future of that configuration are made available. ...

The scale of compression when one adopts the intentional stance toward the two-dimensional chess-playing computer galaxy is stupendous: it is the difference between figuring out in your head what white’s most likely (best) move is versus calculating the state of a few trillion pixels through a few hundred thousand generations. But the scale of the savings is really no greater in the Life world than in our own. Predicting that someone

will duck if you throw a brick at him is easy from the folk-psychological stance; it is and will always be intractable if you have to trace the photons from brick to eyeball, the neurotransmitters from optic nerve to motor nerve, and so forth.

Dennett moves on from this observation to discuss Fodor's position with respect to regularities and whether or not they must be mirrored in the brain. (p. 42)

For Fodor, ... beliefs and their kin would not be real unless the pattern dimly discernible from the perspective of folk psychology could also be discerned (more clearly, with less noise) as a pattern of structures in the brain.

Dennett then returns (p. 43) to discussing regularities in the world. He claims that

Philosophers have tended to ignore a variety of regularity intermediate between the regularities of planets and other objects "obeying" the laws of physics and the regularities of rule-following (that is rule-*consulting*) systems. These intermediate regularities are those which are preserved under selection pressure: the regularities dictated by principles of good design and hence homed in on any self-designing systems. That is, a "rule of thought" may be much more than a mere regularity; it may be a wise rule, a rule one would design a system by if one were a system designer ... Such rules no more need to be explicitly represented than do the principles of aerodynamics that are honored in the design of birds' wings.

It isn't clear to me to which regularities Dennett is referring. Is he really saying that regularities that have been discovered by evolution (or those common to engineering or creative design) have been ignored by philosophers? It doesn't seem to matter, though, because Dennett doesn't discuss these regularities either.

Dennett then returns to relationships between individuals' beliefs, the predictions they may make about the world, and the generally noisy patterns on which those beliefs and predictions are based. (p. 45)

Fodor takes beliefs to be things in the head—just like cells and blood vessels and viruses. ... Churchland [with whom Dennett agrees on this point favors understanding beliefs as] indirect "measurements" of a reality diffused in the behavioral dispositions of the brain (and body). We think beliefs are real enough to call real just so long as belief talk measures these complex behavior-disposing organs as predictively as it does.

Much of the rest of the paper is devoted to arguing that two individuals may see two different patterns in the same data and that (p. 48) "such radical indeterminacy is a genuine and stable possibility."

Dennett allows (p. 49) for the possibility of correctly deciding which of two such competing positions is "correct" by dropping "down from the intentional stance to the design or physical stances." On the other hand, (p. 49) he says that

there could be two different systems of belief attribution to an individual which differed *substantially* in what they attributed—even yielding substantially different predictions of the individual's future behavior—and yet where no deeper fact of the matter could establish that one was a description of the individual's *real* beliefs and the other not. In other words, there could be two different, but equally real, patterns discernible in the noisy world. The rival theorists would not even agree on which parts of the world were pattern and which were noise, and yet nothing deeper would settle the issue. The choice

of a pattern would indeed be up to the observer, a matter to be decided on idiosyncratic pragmatic grounds.

Dennett ends (p. 51) with the following.

A truly general-purpose, robust system of pattern description more valuable than the intentional stance is not an impossibility, but anyone who wants to bet on it might care to talk to me about the odds they will take.

What does all this show? Not that Fodor's industrial-strength Realism must be false, and not that Churchland's eliminative materialism must be false, but just that both views are gratuitously strong forms of materialism—presumptive theses way out in front of the empirical support they require. Rorty's view errs in the opposite direction, ignoring the impressive empirical track record that distinguishes the intentional stance from the astrological stance. Davidson's intermediate position, like mine, ties reality to the brute existence of pattern, but Davidson has overlooked the possibility of two or more *conflicting* patterns being superimposed on the same data—a more radical indeterminacy of translation than he had supposed possible. Now, once again, is the view that I am defending here a sort of instrumentalism or a sort of realism? I think that the view itself is clearer than either of these labels, so I shall leave that question to anyone who still finds illumination in them.

It seems clear that although Dennett must approach some of the issues that arise when exploring questions of reductionism and emergence and that he makes use of the fact that it is possible to emulate a Turing Machine by using Game-of-Life patterns, the focus of "Real patterns" is to offer guidance to philosophers regarding how his views about beliefs are positioned relative to those of other philosophers and not to discuss issues of reductionism, emergence, levels of abstraction, or the reality of higher level entities.

## Reference

Dennett, Daniel C., *The Intentional Stance*, MIT Press, 1987.

Dennett, Daniel C., "Real Patterns," *The Journal of Philosophy*, 1991, (88, 1), 27-51.