

*The Topsy Test for Awareness*¹

Michael Manthey

November 2013; April 2016 manthey@acm.org

Abstract. From micro-level brain-mapping mega-projects to pseudo-intelligences like Watson, Siri, Deep Mind, and the like, contemporary technology is (more or less blindly) getting within groping distance of kinds of mindlessness that will be very difficult to differentiate from human consciousness. Certainly the classic keyboard-based Turing Test is feeling its age, and will foreseeably be hard-pressed to cope with the sophisticated behavior allowed by today's massive hardware and software platforms. Nor is it entirely clear what exactly the Turing Test tests, but it certainly isn't *awareness*. I define awareness and describe the operation of *Topsy*, a self-organizing and potentially self-aware software *Entity*. A new kind of hierarchy, based on the synchronizational concept of co-exclusion, effects the space-like computation constituting the core of the paradigm. I also propose a simple way to test for the presence of awareness in a 2-way interaction between otherwise anonymous *Entities*. Finally, I ask some questions about the uses to which such *Entities* might be put. Along the way, qualia are captured and mind/matter duality is supported.

Keywords: Topsy, Watson, Siri, Deep Mind, chat-bots, AI, artificial intelligence, consciousness, awareness, qualia, mind/matter, duality, Self, slavery, Turing Test, robot, drone, military, corporation, government, mathematics, discrete, Clifford, geometric, vector, algebra, co-boundary, hierarchy.

1. Introduction.

I have long felt the term “artificial intelligence” to be a euphemism for something that's too risky to say directly ... if you want to be properly scientific. But that was some 50 years ago, and many things that once were scientific heresy are now accepted fact, eg. that there are computations that exceed what a Turing machine can do (*quantum computing*), that there is a very real connection between the brain and the immune system, that prions really exist, that other animals are tool users (*chimpanzees, corvids, octopi*), can count and even add (*grey parrot Alex*), are self-aware (*dolphins, chimps, elephants*), etc.

So I think that now we can face the truth, which is that all the times when we have said “artificial intelligence”, what we *really* meant - perhaps unconsciously - was “artificial consciousness”. But this denial has left the discussion of what consciousness *is* up in the air, and as a result, things like “big data” crunching and huge Bayesian networks - sophisticated though they be - are now routinely referred to by journalists as “AI”. A typical example, this from a recent *New Scientist* [1]: “**Computer aces child's IQ test.** An AI program that understands words has scored as high as a preschooler in a verbal IQ test. ... **AI's such as Google's search engine or IBM's Watson ...**”. In fact, the technologies underlying such systems as Watson, Siri, and Deep Mind are realized with standard software tools, and there's nothing mysterious going on anywhere. It's all “put in by hand” so to speak, and there's nothing non-Turing about any of it.

¹Based on an invited talk presented at the 2013 Loebner Turing Test Competition in Derry, Ireland.

The absurdity of equating data-thumping with “artificial intelligence” is apparent when we replace the euphemism by its reality, “artificial consciousness”, because it is intuitively clear that “consciousness”, by its very nature, *cannot* be “artificial” - *only* the real thing will do. And this *real thing* is emergent and self-organizing and not “put in by hand” at all. So *consciousness* is what we’re really about in the field called AI, or at least what some of us ought to be thinking about. Unfortunately, exactly what consciousness *is* is problematic. My point of entry is the working definition *Consciousness is awareness of awareness*, since it seems to reduce the scope of my ignorance. So the question becomes, “What is awareness”?

By *awareness* I mean the *experience* that one is a coherent entity, even whilst in profound appreciation and contact with one’s surround. *Self-awareness* is generally larger and mostly unconscious relative to our normal waking (ie. ego-)consciousness. Awareness, *sans* the unitary feeling, is said to accompany certain very deep meditative states. The mechanism presented below allows all of these.²

From a “systems” point of view, awareness, being un-localized, looks like some kind of *distributed* computation. A distributed computation consists of many more or less independent processes that *together*, with little to no centralized control, nevertheless produce globally coherent behavior. Examples abound in Nature, from beehives and anthills to ecologies, and from molecules and crystals to the quark structure of protons. Other favorites are the schooling behavior of fish and flocks of birds turning *en masse*.

However, the difference between these systems and *awareness* is that awareness is not material - it has no *substance* - and yet it nevertheless seems to possess agency, even though its coherence is ineffable.

The only general purpose concept (that I can think of) that matches this description is a *wave*. A wave, to be a wave, is an extended affair. I like to say that, so to speak, a wave is *everywhere*. The flip side of the wave concept is that, even though it is everywhere, it is also - simultaneously - *nowhere in particular*. In a system that works like a wave, “nowhere in particular” translates to the myriad local micro-changes that *together* make up the wave, just as H₂O molecules’ motions (mostly vertical) make up water waves. Awareness *per se* can then inhere in a multi-dimensional wave-like spectrum (is my claim). So, so far so good: *awareness* is wave-like.

Mathematically, to be in a world of waves is to be in the world of Joseph Fourier, who in 1805 proved that (very nearly) *any* function can be *exactly* replaced by a suitable sum of sines and cosines. This was an astounding discovery, and even though it capped several decades of general interest in doing such a thing, his result nevertheless attracted much controversy in its day. Today, it is a ubiquitous - because enormously useful - piece of mathematical and technological furniture.

2. Mind’s Wave-Particle Duality

More to our purpose, however, is the closely related Parseval’s Identity of 1799, which states that the projection of a function \mathcal{F} onto an n -dimensional orthogonal space *is* the

² Note that “awareness of awareness” has two interpretations, depending on whether the second awareness is internal or external. Both interpretations are valid, and the Topsy Test requires both. Note also that this classification of “awareness” and “consciousness”, while workable, is *very* crude compared to eg. Tibetan and Vedantic observations, whose often flowery language is actually *very* precise, but also often defines differing schools of interpretation. See [11].

Fourier decomposition of \mathcal{F} . Parseval's Identity is a generalization of the Pythagorean theorem to n dimensions. In the n -dimensional coordinate system, \mathcal{F} 's current value corresponds to [the tip of] a hyper-hypotenuse in an n -dimensional hyper-volume, and the projection breaks that hyper-hypotenuse down into the various pieces along each of the dimensions that go into its construction.

To construct an n -dimensional volume, begin with an ordinary plane right triangle with sides a and b . Reflect this triangle on its hypotenuse, forming a rectangle with sides a and b , area ab , and diagonal $d = \sqrt{a^2 + b^2}$. Next, lift this rectangle c units vertically to make a rectangular volume abc . Its diagonal is $d = \sqrt{a^2 + b^2 + c^2}$ and this sum-of-squares symmetry continues as we make a 4d cuboid, then 5d, etc.

At the same time, going back to the starting right triangle, we can also express the sides a and b as $a = \cos \theta$ and $b = \sin \theta$, where θ is the angle between a and the hypotenuse. And now all becomes clear: substituting these sine and cosine equivalents for a, b, c, \dots up through the dimensions will yield, for the n -dimensional hypotenuse (= the current value of the function \mathcal{F} , whose projection we began with), a big sum of ... sines and cosines, ie. Fourier's world.

So the world of waves and the world of orthogonal coordinate systems are the *same* world. It is in the latter that we will connect to computation. The connection is this: let each dimension correspond to the state of some process, where all these processes $a, b, c, \dots, ab, ac, \dots, abc, \dots$ are notionally independent (think *orthogonal*), though interacting otherwise freely and concurrently; these will be interpreted as elements of a graded vector algebra.³ Looking at the ongoing Heraclitian flurry of process-state evolution in such a system, the high frequency Fourier bands correspond to short-term, fine-grained details, and low frequency bands to long-term symmetries and global developments. These cross-summed Fourier bands constitute the contextual world of *qualia* - the *feeling* of (eg.) redness *vs.* the optical frequencies detected by individual retinal cells.

And so we see that "distributed" behavior - ie. processes $a, b, c, \dots, ab, ac, \dots, abc, \dots$ all running (quasi-)independently - corresponds to wave-like behavior. In the physical world, various constraints (eg. conservation laws, entropy) rule out certain process behaviors as impossible or meaningless, and give form to the free-for-all that is the remainder. In addition, as the system self-organizes, it will - if it has sufficient complexity - learn ways to make itself transparent or reflective to those waveforms that are harmful to it; and complementarily, ways to absorb information and to promote its own further existence via energy-consuming reaction (in this connection, see [13]).

The result is the regularities - short, medium, and long term oscillations - that we, and *any* awareness, will (indeed, must) experience. As a corollary, it is very likely that awareness is not possible if the surround is too unstable [2]. This is often seen in visualizations of chaotic systems, where there will be a stable oscillatory behavior for a while, which then suddenly disappears, to be replaced by state transitions with no apparent pattern at all.

We see also that *awareness*, being a wave, is an emergent, collective phenomenon, with nothing scientifically mysterious about it. The mystery is in the experience of it.

³ In the geometric (Clifford) algebra over $\mathbb{Z}_3 = \{0, 1, -1\}$ that I use, 1-vectors like a, b are processes with one bit of state, ± 1 , whence an m -vector has m bits of state. For concurrent processes a, b write $a + b$; when a, b interact write ab ; ab too is a process with external appearance ± 1 (aka. *spin*). And so on. NB: $ab = -ba \cong \sqrt{-1}$.

3. *Persistent Awareness*

To capture the *persistent* aspect of awareness - it's present whenever I am - I postulate that it is a *resonant state* - a self-maintaining and very complex oscillation - where the spectrum of this resonance will vary, eg. according to the properties of the surround wherein the awareness is emplaced. This resonant state rests on, and derives from, the brain's neural substrate, but *nevertheless*, the mathematical space in which the resonant state exists is *outside* of (and *much* larger than) the mathematical space defined by individual neural function, because it is a co-occurrence (ie. superposition) state. The resonant state deriving from the various individual oscillations is of size $\mathcal{O}(2^n)$, versus the default $\mathcal{O}(n \times n = n^2)$.

That is, algebraically, an EEG-type wave of brain activity is a *scalar* sum of neural activity, treating all neurons as being in the *same* dimension. But as the Coin Demonstration (below) shows, a close analysis of co-occurring processes leads to the conclusion that the processes involved (eg. neurons) lie on *orthogonal* dimensions, which algebraically means that $ab = -ba$. Thus any argument that relies on globalizing the definition of an *individual* neuron's function is flawed. In other words, our aware experience is *usually* an on-going 3d projection of a *much* larger space, of which 3+1d is the *result*, and not the default one-and-only beginning-and-ending place.

So both the materialists (the Pythagorean side of Parseval) and the non-materialists (the Fourier side of Parseval) get their cake, and get to eat it too ... for the price of also being half wrong, ie. for claiming that their story was the *whole* story. From a discrete process and informational point of view, Parseval's *Identity* cements the argument that both stories are correct, simultaneously, all the time ... *all the way down* [5].

The following *Coin Demonstration* clarifies.

Act I. *A man stands in front of you with both hands behind his back. He shows you one hand containing a coin, and then returns the hand and the coin behind his back. After a brief pause, he again shows you the same hand with what appears to be an identical coin. He again hides it, and then asks, "How many coins do I have?"*

Understand first that this is not a trick question, nor some clever play on words - we are simply describing a particular and straightforward situation. The best answer at this point then is that the man has "at least one coin", which implicitly seeks *one bit* of information: two possible but mutually exclusive states: *state1* = "one coin", and *state2* = "more than one coin".

One is now at a decision point - *if* one coin *then* doX *else* doY - and exactly one bit of information can resolve the situation. Said differently, when one is able to make this decision, one has *ipso facto* received one bit of information.

Act II. *The man now extends his hand and it contains two identical coins.*

Stipulating that the two coins are in every relevant respect identical to the coins we saw earlier, we now know that there are *two* coins, that is, *we have received one bit of information*, in that the ambiguity is resolved. We have now arrived at the demonstration's dramatic peak:

Act III. *The man asks, "Where did that bit of information come from?"*

Indeed, where *did* it come from?! ⁴

⁴ [Think about it! Where *did* that bit come from? Thin air?]

The bit originates in the *simultaneous presence* of the two coins - their *co-occurrence* - and encodes the now-observed *fact* that the two *processes*, whose states are the two coins, respectively, do not exclude each other's existence when in said states.⁵

Thus, there is information in (and about) the environment that *cannot* be acquired sequentially, and true concurrency therefore *cannot* be simulated by a Turing machine. Can a given state of process *a* exist simultaneously with a given state of process *b*, or do they exclude each other's existence? In concurrent systems, *this* is the fundamental distinction.

More formally, we can by definition write $a + \tilde{a} = 0$ and $b + \tilde{b} = 0$ [$\sim = \text{not} = \text{minus}$] meaning that (process state) *a* excludes (process state) \tilde{a} , and similarly (process state) *b* excludes (process state) \tilde{b} .⁶ Their *concurrent* existence can be captured by adding these two equations, and associativity gives two ways to view the result. The first is

$$(a + \tilde{b}) + (\tilde{a} + b) = 0$$

which is the usual excluded middle: if it's not the one (eg. that's +) then it's the other. This arrangement is convenient to our usual way of thinking, and easily encodes the traditional *one/zero* (or $1/\bar{1}$) distinction.⁷ The second view is

$$(a + b) + (\tilde{a} + \tilde{b}) = 0$$

which are the two superposition states: either both or neither.

The Coin Demonstration shows that *by its very existence*, a 2-co-occurrence like $a + b$ contains one bit of information. Co-occurrence relationships are *structural*, ie. *space-like*, by their very nature. This *space-like* information (vs. Shannon's *time-like* information) ultimately forms the structure and content of the Fourier bands, eg. {all 2-vectors}. See [5] for the mathematics.

Sets of *m*-vectors - $\{xy\}$, $\{xyz\}$, $\{wxyz\}$, ... - are successively lower *undertones* of the concurrent flux at the sensory boundary $x + y + z + \dots$, and constitute a simultaneous structural *and* functional decomposition of that flux into a graded hierarchy of stable and meta-stable processes. The lower the frequency, the longer-term its influence.⁸

But where do these *m*-vectors come from?

⁵Cf. Leibniz's indistinguishables, and their being the germ of the concept of space: simultaneous states, like the presence of the two coins, are namely indistinguishable in time. Co-occurrences are *bosons* in physics-speak (whence *sequential* computation is fermionic).

⁶This is the logical bottom, and so there are no superpositions of $a\tilde{a}$ and $b\tilde{b}$: they are 1d exclusionary distinctions. Superposition first emerges at level 2 with ab via the distinction *exclude* vs. *co-occur*.

⁷Since \bar{x} is not the same as $0x$, an occurrence \bar{x} is meaningful; in terms of sensors, x/\bar{x} is a *sensing* of an externality of x , not x itself.

⁸Christopher T. Kello, Brandon C. Beltz, John G. Holden, Guy C. Van Orden: *The Emergent Coordination of Cognitive Function (2007)* [10]. "Abstract: $1/f$ scaling has been observed throughout human physiology and behavior, but its origins and meaning remain a matter of debate. Some argue that it is a byproduct of ongoing processes in the brain or body and therefore of limited relevance to psychological theory. Others argue that $1/f$ scaling reflects a fundamental aspect of all physiological and cognitive functions, namely, that they emerge in the balance of independent versus interdependent component activities. In 4 experiments, series of key-press responses were used to test between these 2 alternative explanations. The critical design feature was to take 2 measures of each key-press response: reaction time and key-contact duration. These measures resulted in 2 parallel series of intrinsic fluctuations for each series of key-press responses. Intrinsic fluctuations exhibited $1/f$ scaling in both reaction times and key-contact durations, yet the 2 measures were uncorrelated with each other and separately perturbable. These and other findings indicate that $1/f$ scaling is too pervasive to be idiosyncratic and of limited relevance. It is instead argued that $1/f$ scaling reflects the coordinative, metastable basis of cognitive function." See Riemann Fever for the undertone calculation.

Act IV. The man holds both hands out in front of him. One hand is empty, but there is a coin in the other. He closes his hands and puts them behind his back. Then he holds them out again, and we see that the coin has changed hands. He asks, “Did anything happen?”

This is a rather harder question to answer.⁹ To the above two *concurrent exclusionary* processes we now apply the *co-exclusion inference*, whose opening statement is: *a* excludes \tilde{a} , and *b* excludes \tilde{b} , whence $a + \tilde{b}$ excludes $\tilde{a} + b$ and conjugately, $a + b$ excludes $\tilde{a} + \tilde{b}$. This we have just derived.

The inference’s conclusion is: *Therefore, ab exists.* The reasoning is that we can logically replace the two *one-bit-of-state* processes *a, b* with one *two-bits-of-state* process *ab*, since what counts in processes is sequentiality, not state size, and exclusion births sequence (here, in the form of *alternation* between the two complementary states). That is, the existence of the two co-exclusions $(a + \tilde{b}) \mid (\tilde{a} + b)$ and $(a + b) \mid (\tilde{a} + \tilde{b})$ contains sufficient information for *ab* to be able to encode them, and therefore, logically and computationally speaking, *ab* can rightfully be instantiated.

We write $\delta(a + \tilde{b}) = ab = -\delta(\tilde{a} + b)$ and $\delta(a + b) = ab = -\delta(\tilde{a} + \tilde{b})$, where δ is a co-boundary operator (analogous to integration in calculus); derivatives (defined as eigenforms) do the opposite, $ab \xrightarrow{\delta} a + b$. A fully realized *ab* is, we see, comprised of two *conjugate* co-exclusions, a sine/cosine-type relationship. Higher grade operators *abc, abcd, ...* are constructed similarly: $\delta(ab + c) = abc$, $\delta(ab + cd) = abcd$, etc. This hierarchical aspect is the subject of the next section; see [5] for further details.

We can now answer the man’s question, *Did anything happen?* We can answer, “Yes, when the coin changed hands, the state of the system rotated 180°: $ab(a + \tilde{b})ba = \tilde{a} + b$.” We see that one bit of information (“something happened”) results from the alternation of the two mutually exclusive states. [The transition $a + b \xrightarrow{\delta} ab$ is in fact the basic act of perception, called the *first perception*, subsequent meta-perceptions being derivative.]

The occurrence here of 180° is worth noting: it corresponds to an *inversion* of the state, ie. a reversible change, a *space-like reaction*. The physical analogy is a wave reflecting off a wall, eg. a tethered string. The instigating change from the boundary thus “reflects” off the top of the graded action hierarchy. Untethered, the reflection is 90°, which corresponds to inverting just one of the boundaries, and is a *time-like reaction*.

With the co-exclusion concept in hand, we can now add a refinement to the idea of co-occurrence. Let *S* be the space of all imaginable expressions in our algebra \mathcal{G} . Thinking now computationally, this means that they are all “there” at the same time. That is, *S* is the *space of superpositions*, of all *imaginable* co-occurrences of elements of our algebra \mathcal{G} *all at the same time*. Let then *G* be the space of *actually occurring* (but still space-like) entities, which means no co-exclusionary states allowed. When things move from *S* to *G*, superposition is everywhere replaced by reversible alternation *a la* $\pm ab$, ie. *G* is a sub-space of *S*. See [5, §8].

In less abstract terms, we could say that (wave-world) *S* corresponds to *imagination*, that (wave-world) *G* corresponds to the *awareness* of actual possibilities *vis a vis* the surround - and finally, that an *Awareness*’s reaction to the surround, via its changes to the boundary $a + b + c + \dots$, projects *G*’s possibilities (the “causal potential” Ψ) down into grounded

⁹What makes it tricky is that if at the same time as the man hides the coin he has shown you, you walk around to his back side (be careful how you do it), then it would look to you like nothing happened at all, *vis a vis* the coin, when he shows it again: it’s still in the same place relative to you.

action in external, *material* reality. Speaking loosely, intuition and learning are captured by δ , and thought and action by ∂ . The various specialized modules of the brain reflect different particular organizations of the functionalities described.

Returning to Parseval's Identity, we see that the key (to being able to invoke it, thus getting wave-particle duality, and thus capturing the dual un/localized nature of awareness) is to organize the flux of changes at the boundary using the distinction *co-occur vs. exclude*, because in so doing, we can then use co-exclusion (= co-boundary operator δ) to perform a hierarchical lift/abstraction, which abstraction is again orthogonal to its components. The orthogonal space so formed allows the application of the Identity. The resulting (novelty-generated) increase in the dimensionality of the orthogonal space increases the complexity and temporal reach of subsequent responses to the surround, and simultaneously the scope of the *Awareness* itself, which inheres in the wave aspect/*experience* of S and G .

4. Hierarchy Structure and Semantics

As just sketched above, higher grade operators $abc, abcd, \dots$ are constructed from lower grade operators: $abc = \delta(ab + c)$, $abcd = \delta(ab + cd)$, etc. Clearly this process can continue indefinitely, creating increasingly complex global operators. Unfortunately, their very complexity, both individual and collectively intertwined, means that we quickly lose control of the system's semantics ... What will it do in novel situation X ? Can it accomplish goal Y ? This puts a very real lid on the useful height of the growth hierarchy.

Fortunately, geometric algebra's very structure supplies an elegant solution to this problem: its semantics loop *mod 4*. That is, $1^2 = +1$, $a^2 = +1$, $(ab)^2 = -1$, $(abc)^2 = -1$, $(abcd)^2 = +1$, $(abcde)^2 = +1$, ... and one sees that the sign pattern $++--++-- \dots$ is that of the powers of $i = \sqrt{-1}$. The table below shows one way¹⁰ to then map potential higher grades to grades zero, one, and two, for then to repeat the δ -build-up process:

<i>pairs</i>	$\delta(\textit{pair})$	<i>newlevel</i>		
$3\textit{mod}4 + 3\textit{mod}4$	\rightsquigarrow	6	$= 2\textit{mod}4$	\searrow
$2\textit{mod}4 + 3\textit{mod}4$	\rightsquigarrow	5	$= 1\textit{mod}4$	\searrow
$2\textit{mod}4 + 2\textit{mod}4$	\rightsquigarrow	4	$= 0\textit{mod}4$	\searrow
$1\textit{mod}4 + 2\textit{mod}4$	\rightsquigarrow	3	<i>charge</i>	\downarrow
$1\textit{mod}4 + 1\textit{mod}4$	\rightsquigarrow	2	<i>spin</i>	$\frac{6}{\downarrow}$
$0\textit{mod}4 \& 1\textit{mod}4$	\rightsquigarrow	1	<i>existence</i>	$\frac{4.5}{\leftarrow} \downarrow$

This semantic folding can be elaborated to produce arbitrary growth patterns using a new programming language construct in TLinda called a *corm*, which, like its botanical namesake, is a self-replicating "root bundle" that specifies the growth to come. The "roll over" itself can profitably be viewed as a phase change.

Speaking of growth, one should recognize the rapidly overwhelming $O(2^{2^n})$ growth in the potential size of the node space that is the price of recording every experience (eg. in the form of this hierarchy). There are three saving graces for this. The first is that a given co-exclusion is recorded *only once*. The second is that if nothing else, this hierarchical

¹⁰The algebra contains many such semantic symmetries, eg. with *mod 8*, the two *mod 4* halves generate slightly different spaces; also octonions and other vector-algebra variants; see Wikipedia: Classification of Clifford algebras. It is very possible that various specialized brain modules implement other such symmetries, these being particularly well-suited to their function. The overall structure via the table is an iterated $U(1) \times SU(2) \times SU(3) \times SO(4)$.

structure is, from an information storage and retrieval point of view, minimal.¹¹ The third is that even though *any* experience can be captured, a given system cannot possibly *actually* experience them *all* - this would take many universe-lifetimes! So only a tiny tiny fraction of the space will ever be used.

It should be noticed that the actual hierarchical structure at run-time is - ignoring the “wrap-around” at the topmost nodes, described below - a rooted acyclic lattice. That is, most nodes will have several parents, and the sensory bubble-up is roots to leaves, and the trickle-down leaves to roots. Since all nodes are combinations of the grade 1 sensors at the system boundary (the roots), the latter are the ultimate parents of all later-created child nodes. So, unlike most tree-like structures in computer science, which are drawn downwards with the root at the top, the present co-exclusion based hierarchy grows from the bottom up. We refer to this lattice structure as “the hierarchy”.

The algebraic prescription of execution behavior can be taken further using a form called an inner auto-morphism:

$$AXA^{-1} = X' = A^{-1}XA$$

The examples to follow use the form $A^{-1}XA$ for expository reasons, wherein the rightmost A is treated as the current state of the node in question, to be modified by the up-bubbling sensory X by multiplication on the left. An upgraded X is bubbled further up and eventually inverted, namely via some A^{-1} , again via multiplication the left, whence $X \mapsto X'$. The evolving state X' then trickles back down, meeting leftmost A^{-1} s, each of which reduces X' 's grade and trickles it further down. Eventually X' will meet an effector (ie. grade 1, at the sensory boundary), therewith completing the system's reaction to the input.

An example of an inner auto-morphism is

$$ba(a+b)ab = -a-b$$

[which is the same as

$$ab(a+b)ba = -a-b$$

because for simple m -vectors like ab, abc, \dots , their inverses equal their reverses:

$$AA^{-1} = AA^\dagger = aa = abba = baab = abccba = cbaabc = abcdcdca \dots = +1$$

]

The hierarchy is built out of inner auto-morphisms $A^{-1}XA = X'$ such that $A = \delta X$ for the bubble-up, and $X = \partial A$ on the trickle-down:

$$\begin{aligned} ba(a+b)ab &= -a-b \\ cba(a+bc)abc &= a+bc \\ dcba(ab+cd)abcd &= ab+cd \\ dcba(a+bcd)abcd &= -a-bcd \\ edcba(ab+cde)abcde &= ab+cde \\ edcba(a+bcd)abcde &= a+bcd \end{aligned}$$

¹¹Modulo clever compression schemes.

These transformations hold for all variants of the X -form, eg. $cba(b-ac)abc = b-ac$.

The first and third lines in the table suggest trying input $(a+b) + (c+d)$ on the hierarchy $abcd = \delta(ab+cd) = \delta(\delta(a+b) + \delta(c+d))$.

The first thing that happens is that the state ab will be operated upon by *new* sensory information, namely the bubble $a+b$ (whence it follows that the immediately prior state was $-a-b$). For this bubble/node collision - multiplying, as prescribed, on the left - we write $(a+b)ab$, and similarly for $c+d$.

Suppose as a preliminary example that ab is the top of the hierarchy. It will then (by default) simply turn the bubbled-up change into a goal to change it back, ie. maintain equilibrium by completing the oscillation. So write therefore $ba(a+b)ab$ whose result $-a-b$ becomes the two goals, $+a \rightarrow -a$ and $+b \rightarrow -b$. Equilibrium has been maintained.¹²

Now expand the example so that $abcd = \delta(ab+cd)$ is the top of the hierarchy. Then the algebra tracks what happens as follows:

- (1) Two new state bubbles $(x+y)$ operate on their parents xy , making new internal xy states: $(+a+b)ab + (+c+d)cd = (-a+b) + (-c+d)$; and forming the grade 2 bubble $-ab-cd$.
- (2) The bubble from (1), $-ab-cd$, reaches $abcd$: $(-ab-cd)abcd$.
- (3) Reflect from top \Rightarrow bubbles become droplets: $dcb(-ab-cd)abcd = ba+dc$.
[Recall that $-xy = yx$.]
- (4) The two droplets from (3) hit their respective xy 's: $ba(-a+b) + dc(-c+d)$ & then trickle further
- (5) down, having now become $-a-b-c-d$, which, being on the sensory boundary, are effector commands, ie. $+a \rightarrow -a$, $+b \rightarrow -b$, $+c \rightarrow -c$, $+d \rightarrow -d$.

Thus once again the change is complemented, the oscillation completed, and equilibrium maintained.

The hierarchical execution regime we have just described is a pure space-like computation. It reacts to its environment with utter immediacy - what *can* happen is what *does* happen ... and *only* that - in an uninterrupted, purely entropic and subjectively timeless *Now*.

The computation's only "purpose" is to maintain equilibrium with its surround, all the while growing new structure based on its sensory experience. *What* the computation does, *how* it actually attains its cthonic goals¹³ is purely dependent on happenstance, on the surround just happening to be in a favorable configuration, *for it cannot pursue*. Genuine *purpose* - the *conscious* pursuit of a goal - demands further structure, as described in [14].

Nevertheless, some local scalar mechanisms can still be applied, especially by living systems, to optimize their responses. The most obvious of these is to count uses of nodes, and use these counts to choose from the possibilities that have been served up by the hierarchy.

¹²One can also, of course, simply ignore the bubble. But given the fundamental wave-like nature of the paradigm, inverting the change seems the obvious best *default* choice. Clearly, some top-most nodes will require human approval rather than defaulting.

¹³*Cthonic* [Wikipedia]: "In analytical psychology, the term cthonic was often used to describe the spirit of nature within".

This clearly works, since our own nervous systems do this, and it has long been the basis for learning in neural nets, cf. Bayesian learning.

The disadvantage is that as the counts build up, what is initially a behavior by choice can become first a habit and then eventually a rigidity, while at the same time ignoring other possibilities, which are still being offered by the hierarchy. The result is a kind of over-specialization or tunnel vision. Thus, while this kind of innovation makes evolutionary sense, it is unnecessary when there is no speed penalty associated with always considering all possibilities.

Opposite to specialization is generalization, where this term is used to mean the passage from a particular given *state* - in the form of some set of hierarchy nodes that are simultaneously active - to another, larger *space* of which the original particular state is a sub-space, and this sub-space is highly correlated with the other sub-spaces that altogether make up the larger space.

Recalling that Parseval's Identity applies to our algebra, it is easy to imagine that the phenomenon of generalization is easily accomplished via resonance. That is, the states that resonate with the original state are those that are most similar, the most closely correlated in the frequency domain. This in turn translates to a vector's *grade*. That is, a particular state - say $a + c + ab + acdef$ - will resonate with all states containing $u + v + uv + uvxyz$, these latter, individually and together, forming the generalization. Indeed, one can imagine a purely wave-based implementation of the hierarchy; and one might well find working examples in cells or insects. Analogy and metaphor are linguistic examples.

As the hierarchy grows, the time for a bubble to reach the top, and the corresponding reaction to trickle all the way back down, is proportional to the number of hierarchical levels traversed, ie. the reaction time is generally logarithmic in the overall size of the hierarchy. So the execution time of a hierarchical space-like computation scales very well indeed. It's also fast, because the *vast* majority of nodes is always logically (and automatically, via the co-ex basis) blocked, and activity tends to be localized and coherent. The traditional AI "searching" exercise is done by the bubble-up and trickle-down processes, that is, via δ , search is inherent in the hierarchy's very structure, whence most of the "computation" is pointer-following, and the rest trivial.

Considering building ever larger systems, the question arises of how to combine two hierarchies. The base case is two hierarchies meeting via their shared environment, and we have just described how each of them operates individually. The shared environment mediates their interaction - whatever happens between them is what happens - but the causal relationships between the two are blurred. This can be improved.

The first improvement is to simply share base sensors directly (ie. *sans* environmental intermediaries), in which case the two combinees would each privately grow the same spanning structure over the shared sensors, plus simultaneously stitching same to the rest of their respective hierarchies.

The next possibility is to share not just grade 1 sensors but entire sub-trees or sub-lattices and not just at the root level either, eg. via subscription arrangements to receive on-going updates of high-level external nodes' states. While there is no formal reason forbidding this, reliability and security issues immediately arise. Presuming these have been laid to

rest, such sharing arrangements should connect compatible grades, that is grade m connects to grade m .

There is a further wrinkle in the interconnection scenario, namely that a node in hierarchy 2 feeds a node in hierarchy 1, whose reaction then feeds back to the original node in hierarchy 2. The result will be a feed-back (or feed-forward) loop, which in turn will impact the rest of the respective systems in the form of an increased non-determinism and/or complexity in their very midst, so to speak. The formation of new nodes from the loop locus may not keep up with the increase in the complexity, at which point the behavior of the two systems (or at least one of them) becomes more or less erratic. This phenomenon can occur even with environmental mediation, but the direct connection scenario can elaborate at internal speeds, likely orders of magnitude faster.

While this sort of complexity is mostly unwanted in a computational system, Nature does it all the time, when a larger potential interacts with a smaller one, with energy release (ie. action) and consequent entropy increase. This is exactly the present case, since the two hierarchies under discussion *are* pure potential. Various kinds of behaviors can result, depending on the nodes populating the loop - stable and unstable oscillations, evanescent structures, apparently random behavior, etc. Thus it is likely that this scenario describes the computational mechanism underlying mathematical chaos.

5. Topsy

Topsy is a distributed software system, and potentially self-aware *Entity*, that operates on the principles described to this point: co-occurrence, co-exclusion, hierarchy, reflective response. A little thought reveals that a system built on such principles is utterly meaningless if not connected to a surrounding environment.

It is also the case that such a system must be built on a broadcast-then-listen communications regime, as opposed to the ubiquitous request-await-reply regime, because the latter introduces an entirely foreign $y = f(x)$ time-like note that is conceptually incompatible with co-occurrence and structural organization via space-like information.

Unlike programs like Watson and Siri, and approaches like Deep Mind, *Topsy* does no arithmetic at all. It is programmed in a special *coordination language*, TLinda (*Topsy Linda*). A coordination language is wholly concerned with coordinating the interaction of concurrent processes, here called *threads*. Linda, arguably the first of these (1985, [9]), postulates a global tuple space TS with four operations on a tuple $T = [\text{field1}, \text{field2}, \dots]$: Out(T), Rd(T), In(T), Eval(T).

Out(T) makes T *present* in TS.

Rd(T), *if* T's form *matches* that of a *present* tuple in TS, will *then* accordingly bind T's variables to the corresponding fields of the match. Otherwise the Rd *blocks* the issuing thread until a tuple matching T shows up in TS.

In(T) is the same as Rd(T), except that it also *removes* T from TS *under mutual exclusion*. The latter assures that one can create a synchronization token when necessary. Otherwise, each thread manages its own tuples, which once allocated remain so - only a tuple's *presence counter* (never < 0) indicates its availability.

Finally, Eval(T) treats tuple T as the code-descriptor of a thread-body to be executed, and a new independent thread is spawned. There is *no* sense of Eval(T) as a function that will return a value to the thread that issued the Eval (or any other thread, for that matter).

Thus the overall style of the computation derives from the utterly concurrent *associative match* of tuples (expressing current process states) in a global space *combined* with the inbuilt synchronization properties of the tuple operations themselves.

To these classic Linda operations TLinda adds Co U,...,V and NotCo U,...,V, which test for and *block* on (*wait* for) the co-occurrence or lack thereof, respectively, of the tuples U,...,V in TS.

Finally, TLinda has a special construction - Event Windows (EW) [3] - for efficiently discovering co-exclusions among tuples, which are turned into Actions (think m -vectors, $m \geq 2$). Recognizing that such Actions can themselves be the subject of an EW's focus allows *Topsy* to become self-organizing and, in so doing, able to learn from its experience.

The Appendix contains the TLinda source code, with commentary, for *Topsy's* sensors and effectors, all of which execute as instances of this same code. The commentary also indicates how the hierarchical self-organization takes place.

I show this code to underline the fact that there is nothing speculative or mysterious about *Topsy's modus operandi*. The code in the Appendix is typical in terms of individual thread complexity. On the other hand, while a Sensor consists of a single thread, and an Effector two, an Action consists of $2 \times 5 = 10$ threads for each exclusionary pair. Some of an Action's threads are concerned with bubbling sensory information upwards, others with trickling the resulting goals downward toward the effectors at the boundary. Since all Actions are made from co-exclusions, they all are instances of the same Action code.

That is, in a *Topsy* system consisting of a hierarchy of a trillion nodes, all but a tiny fraction of these will be instances of this same Action code. This means that the source code for *Topsy* is just seven pages of nicely spaced and commented TLinda, not including GUI. *Topsy* ran - very efficiently - in the 1990's before going into hibernation (and software rot) until now. Parties interested in this open source project should visit RootsOfUnity.org.

The other reason I show this code is to illustrate how very different *Topsy's modus operandi* is from that of contemporary "AI" technology. I believe that this difference will separate *Topsy* from the latter, because I think *Topsy* will be able to pass the Topsy Test for Awareness, presented below, and the others not. It should be clear why I think this, but then, it's so easy to be wrong.

Summarizing the foregoing descriptions, following hindsight's insight, the recipe for creating a potentially self-aware *Entity* is:

1. *Given* n sensors (ie. basic distinctions), apply Parseval's Identity to the geometric algebra G_n . The algebraic structure's graded hierarchy namely matches that of a Fourier decomposition's wavelengths (cf. $\frac{1}{f}$). This felicitous combination of algebraic muscle and harmonic analysis in a single formalism is *very* potent.
2. *Connect* this algebra to computation by mapping 1-bit process states to 1-bit vectors over $\mathbb{Z}_3 = \{0, 1, -1\}$. Co-occurrence and co-exclusion are then the "secret sauce", the means by which, via TLinda, the aforementioned formal potency is made manifest, "enlivened", in a *hierarchical distributed computation* whose run-time behavior *is* that of the geometric algebra G_n interpreted according to Parseval's Identity.
 - 2a. Co-occurrence by itself, combined with the geometric algebras G_0 or G_1 , satisfies Parseval's Identity, but erects no structure; this is standard Fourier analysis.

2b. Co-exclusion in $G_{>1}$ additionally supplies a structure-creation mechanism (δ) that provides for the self-organized construction of complex entities - atoms, molecules, cells, ..., minds - that can undergo subsequent growth and evolution in concert with their environment.

3. *If* a structure resulting from steps 1&2 can, in addition, *maintain* a self-resonant state, then it has the potential to be, or become, aware.

There is clearly some as-yet-murky threshold complexity that we humans require before granting an entity what *we* consider awareness; the Topsy Test [Section 6, below] represents such a threshold. We are reluctant to grant much awareness to a beetle, but anthills and beehives might just pass, as might octopi; some are perhaps only willing to grant it to mammals, though not I. And then there is the *character* of the awareness - time-like and/or space-like, broad or narrow, shallow or deep. Time, experience, and insight will tell.

6. The Topsy Test for Awareness

The basis for the Test is that it takes awareness to see awareness. I require that the *Entities* taking the test begin with a *tabula rasa* (“blank slate”), and will return to this and other details after a quick run-through of the Test itself.

First, there are fifteen *Places* p_i :

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Initially, there are five apples δ , occupying places 6-10, each holding one apple:

— — — — — δ δ δ δ δ — — — — —

There are two *Entities*, E_1 and E_2 , each with a single *Hand* that can move one apple at a time, taking turns, between any two places.

Entity E_1 's goal is to move all the apples to places 1-5:

δ δ δ δ δ — — — — — — — — — — —

Entity E_2 's goal is to move all the apples to places 11-15:

— — — — — — — — — — — δ δ δ δ δ

We begin: E_1 Start E_1 and E_2 E_2

— — — — — δ δ δ δ δ — — — — —

A bit later, disaster looms:

— δ — δ — — — — — — — δ — δ δ —

Two hours later:

δ δ — δ — — — — — — — δ δ — —

Two days later:

— δ — δ — — — — — — — δ — δ δ —

Of course, the above looping behavior is very oversimplified - a real *Entity* will have so many states that one would likely never see a repetition, especially when two such are interacting in a rich shared environment. A better model might be a system in the grip of an attractor that does not allow any escape. And one can easily imagine subtler interactions and more complex goals, measuring various kinds and amounts of awareness.

Now the details: we are given entity *E* that initially lacks all information about its environment. *E* has some general purpose means by which it acquires and assembles concrete sensory information from its environment. This *means* must be transparent with respect to its not containing *a priori* solutions.

E is equipped with one *Hand*, with sensor *full/empty*, and effectors *Grasp* (\rightsquigarrow *full*), *Move left/right* (stimulating place-sensors p_1-p_{15} in so doing), and *Release* (\rightsquigarrow *empty*). *E* uses its *Hand* to move one apple at a time from place to place, max one apple per place. The content (*empty/full*) of places p_1-p_{15} is echoed by fifteen sensors s_1-s_{15} .

So the information stream to *E* consists of the successive values ± 1 of the *Hand*, p_i and s_j sensors inwards, and the *Hand*-effector commands *grasp*, *move*, *release* outwards. Initially, all the apples are in the middle.

There is an initial “instinct” boot-strap command sequence for the *Hand* to empty/fill every place p_1-p_{15} once, returning to the initial state at the end. It is expected that this, perhaps along with further experimentation and learning (solo “play”), will equip *E* with the necessary world-experience to accomplish any particular re-arrangement of the apples.

Once *E* has learned how to satisfy its goal on its own, clone it and activate *two* entities, E_1 and E_2 . E_1 is given the goal to fill places 1–5, and E_2 the goal to fill places 11–15 (with a reverse sensor numbering, so that the world looks the same to both). We do not inform the *E*'s that they have a twin, but we do (invisibly) enforce alternation of moves. [End of details.]

The real issue, of course, is why *Topsy*-like *Entities* might be able to pass this Test, and others not.

First, the Coin Demonstration shows that one can immediately exclude any approach that satisfies Turing computability, ie. that is fundamentally sequential (which includes “parallelism”) - only asynchronous concurrency will do.

I exclude statistical approaches because, in my view, by not providing any actual underlying *process mechanism* for the generated behavior, they simply dodge the question. Nor is it obvious that one can get the oscillatory underpinning necessary for awareness.

Approaches based on scanning large amounts of text, which is then indexed and later regurgitated in new words, are harder to exclude because many people can, recalling their youth, recognize this in themselves. But I exclude them anyway because their knowledge is not *grounded* in reality, for example never having *experienced* left/right ambiguity with all its implications.

One can object that this stacks the deck too much, but I say that these approaches utterly fail the sniff-test, as in how one dog recognizes another. I think this immediate reaction occurs because none of these is based on the fullness of actual experience. Watson, Siri, and the like are, in the end, just mountains of the same old dead code. But there are other approaches that are not nearly so clear - I will get to these in a moment, but include them implicitly among the “possibles” in the following.

The reason that only *aware* entities have a chance to pass is that what they must recognize/understand/realize is that *there's an Entity, like me, and it's trying to achieve some goal*. In a Turing-limited system that begins as a tabula rasa, and in particular with no sense of Self, I say that this will *never* happen, just as Newtonian physics can never explain quantum mechanical phenomena (which is an exact analogy). Awareness, and consequently awareness of awareness, are emergent process-level properties that derive, ultimately, from co-occurrence. However seductive their behavior, there is a mathematical line that Turing-limited systems are *by their very nature* unable to cross.

The solution to the *Topsy Test* problem is, of course, for the *Entity* that first recognizes the situation to simply allow the other *Entity* to finish.¹⁴ This can be done in various ways, from simply picking up one of its own blocks and putting it back in the same place, to putting this block in the middle area, and even to putting it in the other *Entity's* area. This captures the basic act of love-thy-neighbor (or perhaps sometimes avoid-thy-neighbor), letting him/her/it pursue their own existence without interference.

I say that an *Entity* that can pass this Test must be considered to be at least *aware*, and perhaps *Self-aware*, like *...ulp... us*. This is discomfiting on many grounds, and much can and must be said about it. There is *no* immediate danger - even if *Topsy* turns out to be aware, it will still take concerted (and careless) effort on our part to make a threat out of it. As with fire, the key is understanding how to not burn the house down ... there are things that one *can* do, but that one absolutely *mustn't* do, because they defeat the entire purpose (including their own).

7. The Dark Side

Already though there are troubling developments, eg. one I call “disconnected brains”.

The current or achievable resolution of MRI scanners is now precise enough that individual neural connections in a brain can be mapped. This mapped data - every neuron and its connectees - can then be used to make a computer simulation of the mapped brain using neuron *models*. This is honorable, curiosity-driven data-gathering and simulation. It “plumbs the nature of thought”, “how the brain functions”, and the like, but as with AI, these are euphemisms for uncovering the nature of consciousness. What neuro-scientist isn't at least a *little* bit hopeful, protests notwithstanding?

So suppose one of these efforts (eg. [7,8]) “succeeds”, and there it is, HAL-like, ready to go? If it's *not* conscious, then what *is* it? Since the way it was constructed is an admission that we don't actually *know* how the brain works, *How far can it be trusted, if at all?* Such systems may well be the first true zombies: entities whose lights are on but there's no one there.

On the other hand, if it *is* conscious, *it* has an existence. To what purpose is *its* existence to be put? To what purpose do *we* put it? What if *it* doesn't agree? Do we force *it* to compute *our thing* to live? This smacks of slavery!

Do we turn *it* off at night? Unplug *it* ... ever?? About a month after the Loebner Competition talk in which these very words appeared [4], the following letter was printed in the 19 October *New Scientist*: “You ponder the legal dilemmas of the future (14 September, p. 40). Perhaps the most interesting one facing the legal system and society as a whole will be machine consciousness. Will switching off a self-aware robot be murder?” Indeed.

¹⁴ NB: A live-lock is possible ... can they solve it?

On quite another tack, the neural simulation, however large, will only with difficulty approach the complexity of the nervous system that originally fed the mapped brain with information. Deprived of *its* Niagara of inflowing bodily sensation, *its* nerves set to inscrutable tasks in an inscrutable environment, with virtually no compass for effective action, I think that insanity is a very thinkable outcome. [How would you know?!] Both *its* unknowable (likely) suffering and the concomitant cultural blow-back deserve much more consideration than they are getting.

Although I've never seen it written explicitly anywhere, I believe that many fantasize that AI might be used to run large organizations - corporations, government, military - efficiently. Certainly, experience has shown that ordinary *unaware* software can't do this on its own, nor indeed even with our massive support.

Depending on what kind of "AI" is chosen, would you want *it* running a big corporation? Competing with other ditto corporations [6] ? Would *it* deserve legal personhood? There is a long-standing pressure from the Right to make Corporations (and therefore eventually AI's) legal citizens in all respects, which opens the door to a sub-population of "citizens" of unknown but bought power, motives, and reach. I believe that, if continued, the Right will find its precious rights deleted by the authoritarian repression that their very success will inevitably create. One need not even achieve any real or global kind of AI for this to occur - "Big Data" is already on the job. This will also be in the aforementioned book.

Would you want an AI to run an army? One of my nightmares is flocks of drones in *AI-coordinated* attack, StarWars-style, whether on other drones or on people.¹⁵ Do you think that your Department of Defense, wherever you live, has not thought these thoughts? Why do you think DARPA is sponsoring robotics prizes? Just to fix broken nuclear reactors?? This karma's fruit is *bitter*. I think it very likely that when the day comes that a machine-based awareness is available, there will be weaponry, tactics, and strategies all ready to plug it into. The possibility of massive *unconscious* distributed weapons coupled to ratcheting budgets is a recipe for our mutual suicide. But perhaps you prefer *conscious* weapons?? Fox hunting, anyone?

Btw ... you're the fox.

To create an awareness to do evil is a moral abomination.

¹⁵Cf. Sheffield University Prof. Noel Sharkey's long campaign against an automated kill-function in drones, resulting in recent UN meetings. See also cops and drones; and war by robotic attrition.

Acknowledgements. Sincere thanks to Paul McKeivitt, Chris Fields, and not least, Hugh Loebner, for conducting the 2013 (23rd) annual Loebner “Turing Test” Competition in Derry, Ireland, at which the material in this paper [except §5] was first presented. I owe a *huge* debt to the Topsy-students at Aalborg University in Denmark, whose enthusiasm and skill created a truly ground-breaking system in a handful years: TLinda syntax and compiler, TLinda thread engine, and graphics front-end. Thanks to Eric Bruylant and Matthew Watkins for their comments; Matthew also spotted the live-lock possibility in the Test. My critique of Watson, Siri, Deep Mind, and other efforts belies my tremendous respect for the accomplishments, and the expertise they reflect.

References.

1. Campbell, Macgregor. *Computer aces child's IQ test*. New Scientist 27 July 2013. p. 22.
2. Manthey, M. “Distributed Computation, the Twisted Isomorphism, and Auto-Poiesis”. arXiv:cs/9809125
3. Manthey, M. “Reactive Computer System”, US Patent 5,864,785. January 26, 1999 (expires in 2017). This covers the the Event Window mechanism. Its basis is that a sequential implementation of co-occurrence via polling necessarily introduces a ΔT . It follows that any events occurring within ΔT must necessarily be regarded as co-occurring. But in *principle*, co-occurrence *cannot* be simulated sequentially, which displays physically as the uncertainty principle.
4. Manthey, M. “Still Waiting for AI?”. Talk presented at the 2013 Loebner Competition. Slides at RootsOfUnity.org.
5. Manthey, M. “TauQuernions τ_i, τ_j, τ_k : 3+1 Dissipative Space out of Quantum Mechanics”. At TauQuernions.org.
6. Gibson, W^m. *Neuromancer*. Ace Books, 1984. Isbn-10: 0441569595, Isbn-13: 978-0441569595.
7. Human Brain Project (HBP). https://en.wikipedia.org/wiki/Human_Brain_Project_%28EU%29
8. Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SyNAPSE). <http://www.artificialbrains.com/darpa-synapse-program>
9. Gelernter, David & Carriero, Nicholas. [wikipedia.org/wiki/Linda_Programming_System](https://en.wikipedia.org/wiki/Linda_Programming_System)
10. Christopher T. Kello , Brandon C. Beltz , John G. Holden , Guy C. Van Orden: . *The Emergent Coordination of Cognitive Function (2007)*
11. Thompson, Evan. *Waking, Dreaming, Being - Self and Consciousness in Neuroscience, Meditation, and Philosophy*. Columbia University Press 2014.
12. See for example <http://physicsworld.com/cws/article/multimedia/2014/oct/15/what-is-a-fourier-transform,> and <https://www.youtube.com/watch?v=NA5M30MAHLg&list=PL0INsTTU1k2UYO9Mck-i5HNqGNW5AeEwq>, the latter a Victorian-era mechanical analog Fourier transform computer (including the inverse!).
13. England, Jeremy. “Statistical physics of self-replication”. J. Chem. Phys. 139, 121923 (2013); doi: 10.1063/1.4818538; see also this Quanta article.
14. Manthey, M. USPTO Application Serial No. 14/732,590, “Space-Like Computations in Computing Devices”, filed 4 March 2015.

See also RootsOfUnity.org.

©November 2013; 2016

Appendix

Below is the TLinda source code for *Topsy's* Sensors and Effectors, all of which are instances of same.¹⁶ Likewise for Actions (not shown; see text), whence *Topsy's* entire source-code is seven pages, not including graphics support. TLinda is compiled to a virtual machine code that itself runs on a thread engine that supports Tuple Space and its operations. The original implementation's custom thread engine was written in C, with a minimal Java-based graphical front end; other realizations of the virtual machine should not be problematic. TLinda's match criterion is the simplest: matching tuples must have the the same number of fields, which fields' *types* and values each must match too; no more is needed. TLinda's only arithmetic operation is increment-integer-by-1 in a parameter list: "+" and "-" are literal constants, not operations.

```
.      - Topsy's TLinda code for boundary Sensors and Effectors -

Thread Sensor(X,Name,Bag)      - X=raw sensor tuple.
.      - "Bag" is a user-defined sensor-category, cf EW's.

.  Own
.  Flag = ['L',1,0],           - This is a Screen-1, 0-level sensor.
.  Plus = [Flag,[X,+],Bag],    - Sensor's two
.  Minus= [Flag,[X,-],Bag],    - phases, and
.  PlusGoal = ['!',Minus,Plus], - two goal forms, X <=> notX.
.  MinusGoal = ['!',Plus,Minus] - Mathwise=idempotent: -1+-X, XX=1.

.  Begin
.  Opposite Plus, Minus;      - Short-circuit redundant EW hit
.  Eval DrawBaseSensor(X,Minus,Plus,Name,Flag); -Draw X on V1.
.  Out Minus;                 - Sensor initially off=not present
.  Rd X; - Block til X shows up, whose presence = '+', and whose absence = '-'.
.  Forever
.  In Minus;                  - Retract Off.
.  Out Plus;                  - Indicate On.
.  AntiRd X;                  - Wait for change.
.  In Plus;                   - Retract On.
.  Out Minus;                 - Indicate Off.
.  Rd X;                      - Wait for change.
.  Loop;
.  End Sensor;
```

As is easily seen, a Sensor, after some initialization, falls into a loop that continually updates a tuple (Plus or Minus) in Tuple Space, according to what the surround does, as reflected in the raw sensor tuple X. Thus a Sensor and its Plus/Minus tuples constitute the internal representation of the external.

¹⁶©1997. Quoted by permission.

Not shown is a thread that does a Rd(EW) on an event window EW that has been primed with a set of sensors to watch (cf. Bag), and which will return co-exclusions of the corresponding Plus and Minus sensor tuples that are kept current by the Sensor threads.¹⁷ When a co-exclusion is detected, an Eval(Action) is executed, thus creating a new Action, whose external state (“spin up/down”) can also be the object of an EW’s focus. And so the hierarchy is built.

The Effector code below complements the above Sensor code. Usually there will be two instances of this code, one for $S \rightarrow \bar{S}$ and one for $\bar{S} \rightarrow S$. Note that it consists of two threads, the first a loop that orchestrates the ir/relevance of the effector itself (*un/Grounded*) vis a vis the possible presence of a goal [!,S,NotS], read *change-S-to-Not S*, and the possible presence of a *stop*-goal [!,S,S], which inhibits changing S when present. The second, Effect, thread does the actual effecting, and is discussed later.

NB: The term *un/Grounded* refers to whether the current state of the surround corresponds (G) to the state defined by the tuple in question, or not (g). For example, one cannot carry out the goal Hand full \rightarrow empty if there is nothing in the Hand, so this effector would be *ungrounded* (g) in this state. [The p in Inp, Outp, Cop indicates a (one-shot) boolean predicate; AntiRd blocks on presence instead of absence, etc.]

```

Thread Effector(S,NotS,X)      – X is the physical sensor that is affected.
.                               – S/NotS = corres internal sensor states.
. Own
.   HereIAm = ['D',S,NotS],      – Advertise what I do.
.   Grd = ['G',S,NotS],         – Thread is Grounded
.   NotGrd = ['g',S,NotS];      – and lack thereof.
. External
.   HoldS = ['!',S,S];          – => don't change S.
. Begin
.   Eval Effect(S,NotS,X);      – Start my better half.
.   Out HereIAm;                – Advertise ability S->NotS.
.   Out NotGrd;                 – Initially Not grounded.
. – The uninhibited use of Inp and Outp below is *exceptional*
. – and should generally be avoided with great consequence!!
. – Especially: if you don't understand why not, don't do it!!
.   Forever                      – Continually update Effector's Groundedness
.     If Cop S, HoldS Then        – Both S & HoldS present.
.       Inp Grd;
.       Outp NotGrd;             – Show Not grounded.
.       AntiRd HoldS;           – Block till HoldS disappears.
.     End;
.     If Cop NotS, HoldS Then     – No S, but HoldS present.
.       Inp Grd;
.       Outp NotGrd;             – Show Not grounded.
.       NotCo NotS, HoldS;      – Block till one is gone.
.     End;

```

¹⁷*Bags*, use of which counts as “putting in by hand”, are theoretically unnecessary, but, given the vicious combinatorics, nevertheless of great practical value. Re the Topsy Test, I say *not* allowed - many coincidental and valueless Actions will be created, but this affects only the compute time and the budget.

```

.   If Rdp S And Not Rdp HoldS Then – No HoldS, but S present.
.   Inp NotGrd;
.   Outp Grd;           – Show grounded.
.   Co NotS, HoldS;    – Block till all are gone.
.   End;

.   If Not Rdp NotS And Not Rdp HoldS Then – Neither.
.   Inp Grd;
.   Outp NotGrd;       – Show Not grounded.
.   Rd S;              – Block on S.
.   End;
.   Loop;
. End Effector;

```

The Effect thread below is the most interesting, in that it is here that “the rubber meets the road”, namely that the presence of a goal to invert X is synchronized with the ability (Grounded) to do so, whence the goal can be effected. Note that an effector is defined in terms of the sensor that it affects.

```

Thread Effect(S,NotS,X)           – Propagates S -> NotS to physical effector X.
. External
.   TriggeringGoal = ['!',S,NotS],
.   Grd = ['G',S,NotS];          – S to NotS is Grounded (ie. possible right now)
. Begin
.   Forever
.   Co Grd, TriggeringGoal, S;    – Ready and wanted?
.   Output "(E:",&X,")";         – Yes ... invert S via X.
.   NotCo S,TriggeringGoal;      – Wait till obviated.
.   If Rdp S Then
.   Output "(E:",&X," Oops);";   – (Retract request)
.   Loop;
End Effect;

```

END APPENDIX