★ FORUM ★

A Turing Program for Linguistic Theory

Jeffrey Watumull

1. The Big Question

Universal to systems so various and complex as the foundations of mathematics, cryptography, computer science, artificial intelligence, and morphogenesis is the "the concept of 'mechanical procedure' (alias 'algorithm' or 'computation procedure' or 'finite combinatorial procedure'). This concept is shown to be equivalent to that of a 'Turing machine.' [In fact,] due to A.M. Turing's work, a precise and unquestionably adequate definition of the general concept of formal system can now be given" (Gödel in Davis 1965: 71–72,). For this triumph, *inter alia*, the world is now celebrating the centenary of the mathematician Alan Mathison Turing (1912–1954).[1] The mathematical universality of the Turing machine — the abstract system of computation — implies that it is not only *relevant* to biolinguistic research, but *intrinsic* to linguistic — indeed any cognitive-neurobiological — computation. To demonstrate this is the desideratum of my proposed *Turing Program for Linguistic Theory* (TPLT). The proposal is very summary and very sketchy; I proffer no answers, only approaches to questions.

One of the "Big Questions" of the Turing Centenary (see http://www.turingcentenary.eu) is whether there exists "a successful mathematical model of intelligent thought". The answer is surely not yet, but I am sure there will be. Cognition is clearly computational (see Gallistel & King 2009) and computation is mathematical by definition: Procedures are run to determine the symbolic outputs (values) of functions given symbolic inputs (arguments); in the domain of the brain, the running of procedures is referred to informally as 'thinking'. In a successful model of this process, functions would be "*completely* determined" (Turing 1936: 232, emphasis original) by rules and representations so "perfectly explicit" (Chomsky 1965: 4) as to be automatable.[2]

For comments and criticisms, my many thanks to Bob Berwick, Noam Chomsky, Randy Gallistel, Marc Hauser, Steve Pinker, and Ian Roberts.

[1] From the editors of the "Turing at 100" special issue of *Nature* (2012, 482: 440):

> *Nature* invites its readers to embrace and celebrate [Turing,] one of the brightest minds of all time […]. The scope of Turing's achievements is extraordinary. Mathematicians will honour the man who cracked David Hilbert's *Entscheidungsproblem* or 'decision problem,' and cryptographers and historians will remember him as the man who broke Nazi Germany's Enigma code and helped to shorten the Second World War. Engineers will hail the founder of the digital age and artificial intelligence. Biologists will pay homage to the theoretician of morphogenesis, and physicists will raise a glass to the pioneer of nonlinear dynamics.

[2] For "*if you can't program it, you haven't understood it*" (Deutsch 2011: 146, emphasis original).

ISSN 1450–3417

The successful model would be *descriptively* and *explanatorily adequate* (see Chomsky 1964): It would completely describe *what* the system of rules and representations is (see Roberts 2011) and completely explain *how* it develops — how internal and external factors determine the genotype-to-phenotype expression of the cognitive system (see Pinker 1984). The model might even go *beyond* explanatory adequacy (see Chomsky 2004a) to answer the biggest question of all: *Why* does the system assume this one form out of the infinity of conceivable forms? The answer might ultimately derive from computational constraints reducible to mathematical laws. This model would thus succeed in explaining not only *something of the specific nature of intelligence*, but ultimately *something of the general nature of reality*.[3]

These *what*, *how*, and *why* questions are obviously too big to answer as framed, and must therefore be decomposed into smaller solvable problems. Hence I propose the TPLT, a research program based on Turing's mathematics, to discover and model mathematically important aspects of intelligent thought in the domain of language.

2. Language

One strategy for answering the general question whether there could be a successful mathematical model of intelligent thought is to reformulate it as a mathematical modeling problem for a specific form of human intelligence. The form I propose to consider is that of language.

> Why study language? [One reason is to] discover abstract principles that govern its structure and use, principles that are universal by biological necessity and not mere historical accident, that derive from mental characteristics of the species […]. By studying the properties of natural languages, their structure, organization, and use, we may hope to gain some understanding of the specific characteristics of human intelligence. We may hope to learn something about human nature; something significant, if it is true that human cognitive capacity is the truly distinctive and most remarkable characteristic of the species. (Chomsky 1975: 4)

As an explanandum of scientific inquiry, 'intelligence' is so polysemous that any theory not formalizing it is "too meaningless to deserve discussion" (Turing 1950: 442).[4,5] *Mutatis mutandis* for 'language', commonly conflated with

---

[3] I am but one of many to have conjectured that fundamentally the universe *is* mathematical (computational/informational).

[4] Polysemy is not the only (or even the main) problem. As Noam Chomsky (p.c.) observes, the question whether a machine *really* thinks is equally meaningless to the question whether a submarine *really* swims: "These questions have to do with choice of metaphor. They are not substantive. Questions about swimming, thinking, [etc.] are about the meanings of linguistic elements/concepts, and how they are used" (see Chomsky 2009).

[5] Essentialism is comparably meaningless: "[S]ooner or later we will be able to assemble programs of great problem-solving ability from complex combinations of heuristic devices — multiple optimizers, pattern-recognition tricks, planning algebras, recursive administration procedures, and the like. In no one of these will we find the seat of intelligence. Should we ask what intelligence 'really is'?" (Minsky 1963: 446–447). Like an onion — that imperishable if slightly stale staple of analogies — we can peel away the purely mechanical functions of the mind in search of the 'real' mind. If "we eventually come to the skin which

any system of communication or representation (perhaps quasi-symbolic); or interpreted as a social-political construct governed by shared norms and conventions of usage; or classified as "the totality of utterances made in a speech community" (Bloomfield 1926: 155); or reduced to a Skinnerian repertoire of habits, abilities, and dispositions to respond to verbal stimuli; or stipulated to be a set of essentially Quinean well-formed formulae; or obfuscated in a Wittgensteinian game or "form of life". By contrast, modern linguistics — *generative linguistics* (subsuming *biolinguistics*) — established and expounded by Noam Chomsky, in a Turing-style consilience of the formal and natural sciences, adopts a rigorous and empirical definition of 'language' as an *I-language*: A cognitive computational system — a function in *intension* — *internal* to an *individual* of the species *Homo sapiens sapiens*.[6,7] The function recursively generates syntactic structures mappable via formal semantics and rule-based morphology–phonology to interfaces with conceptual-intentional and sensory-motor systems, respectively. I-language is thus a system of *d-infinity* (discrete/denumerable/digital infinity) analogous to the natural numbers: A finite system that in principle can generate an infinite set of hierarchically structured expressions by recursively combining discrete elements.

A generative system *strongly generates* structures/sets (material to linguistic cognition) and *weakly generates* strings/sequences (marginal to linguistic cognition). The structures, not the strings, represent the grammatical information mappable to representations of semantic and morphological-phonological information, as evidenced by the fact that one string can correspond to many structures (in a many-one function). Consider that the one string *the boy saw the man with binoculars* is two-ways ambiguous because it corresponds to two possible structures representing two possible interpretations: (i) {{*the*, *boy*}, {*saw*, {*the*, {*man*, {*with*, *binoculars*}}}}}; (ii) {{*the*, *boy*}, {{*saw*, {*the*, *man*}}, {*with binoculars*}}}.

The strong generative capacity of the language faculty was probably exapted in human evolution to connect interfaces with systems necessary for general intelligence (see Hauser 2009).[8] For instance, Minsky (1963) argues that the sophistication of human pattern-recognition necessitates "provisions for recursive, or at least hierarchical use of previous results" as in the "articulation" of a scene into descriptions of "elementary figures" and "subexpressions […] designating complex subfigures" with a "figure […] first divided into two parts; [and] then [with] each part […] described using the same machinery" (pp. 434, 423).

---

has nothing in it", we cannot but conclude that "the whole mind is mechanical" (Turing 1950: 454–455).

6 To define the function in *extension* is to define the set of syntactic objects (strings/structures) it generates (see Church 1941). For linguistic theory, the function needs to be defined in intension: The definition of the function qua procedure for generating sets of structures; the properties of the structures derive, in part, from the properties of the function.

7 Of course, a universal — species-typical — I-language can be abstracted from particular I-languages for formal and empirical inquiry. And if Platonically inclined, we may abstract a universal I-language as a mathematical object for metaphysical analysis.

8 Updated with technology from generative linguistics, the general problem solving program T.O.T.E. of Miller, Galanter, and Pribram (1960) has been argued to be active in linguistic and extralinguistic domains (see Jackendoff 2007).

And as McCarthy (1956) explained, recursively hierarchical generativity is necessary for complex planning methods because inefficient if not technically intractable problems can be solved only by decomposition into tractable and efficient sub-problems; composition of the functions and solutions for the latter can solve the former for infinite sets of problems.[9] Accordingly, recursively generated "Hierarchy […] is one of the central structural schemes that the architect of complexity uses" (Simon 1962: 468) (see Turing 1947 on the central importance of 'subsidiary tables', i.e. recursive subroutines).

3. Language and the Brain

I submit that the object of linguistic inquiry is, or can be regarded as, "the thing in itself", a computational — ergo mathematical — system abstracted away from spatiotemporal contingencies, as a Turing machine is with its memory space and operating time unlimited: "With this will come a mathematical characterization of a class of […] functions, the functions 'computed' by these Turing machines. These functions will be called *computable functions*, [identifiable with] the intuitive concept of effectively calculable function" (Davis 1958: 3). In short, "[s]omething is computable if it can be computed by a Turing machine" (Gallistel & King 2009: 105)[10] and "[a]ny Turing machine is completely described by a *machine table*" (Putnam 1975: 365), a *functional organization*, specifying its mathematical-logical rules and representations, not its physical implementation (if it even has one): "[T]he 'logical description' of a Turing machine does not include any specification of the *physical nature* of [its rules and representations] — or indeed, of the physical nature of the whole machine […]. In other words, a given 'Turing machine' is an *abstract* machine which may be physically realized in an almost infinite number of different ways" (Putnam 1975: 371, emphases original), *if at all*.[11] So it is mere "superstition" to attach "[i]mportance […] to the fact that modern digital computers are electrical, and that the nervous system also is electrical. Since Babbage's machine was not electrical, and since all digital computers are in a sense equivalent, we see that this use of electricity cannot be of theoretical importance […]. If we wish to find […] similarities, we should look rather for *mathematical analogies of function*" (Turing 1950: 439, emphasis added).[12]

Consistent with this reasoning, the object of linguistic inquiry can be defined as a form of *mathematical-functional competence*: the "underlying system of

---

[9] Behaviorist psychology assumed nonhierarchical chaining theories (Markov models), but these were determined to be inadequate by Lashley (1951) in a paper (unappreciated until Chomsky 1959b) on the necessity of hierarchical planning in solving the problem of serially-ordered behavior.

[10] In this proposal I do not address the limits of the computable (e.g., the relation of artificial neural networks, so-called 'Super-Turing machines', etc. to classical Turing machines). That is a task for the TPLT.

[11] The general and necessary and sufficient conditions an object must satisfy for it to be defined as a (type of) Turing machine — a computer — are purely mathematical-functional (see Carnap 1955 on such intensional definitions).

[12] Charles Babbage (1791–1871) was a Cambridge mathematician and designer of the Difference Engine and the Analytical Engine, which "had all the essential ideas" (i.e. the mathematical-functional components and procedures) of Turing's computers (Turing 1950: 439), but were mechanical.

rules" (Chomsky 1965: 4) in the mind that "represents the information concerning sentence structure that is available, in principle, to one who has acquired the language" (Chomsky 1963: 326–327). This information is represented as an "idealization […] leaving out any limitations […] of memory, time, and access" (Chomsky 1965: 4, 10). Idealization of the linguistic system reveals the mathematical-functional components and procedures necessary and sufficient to define it as a subtype of Turing machine. Such idealization is part and parcel of the methodology and the *metaphysics* of normal science, which proceeds by the "making of abstract mathematical models of the universe to which at least the physicists give *a higher degree of reality* than they accord the ordinary world of sensation" (Weinberg 1976: 28, emphasis added).[13] For those working in the Turing Program for Linguistic Theory (TPLT), it would be natural to give the highest degree of reality to the mathematical form of I-language.

In the "world of sensation", things in themselves, often abstract, are confounded by arbitrary constraints, often physical. For computational systems, confounding the abstract with the physical can conflate the truistic yet only lip serviced distinction between software and hardware; thus this important distinction remains unassimilated, preventing recognition of the fact that "[a]s our knowledge increases, the abstract mathematical world becomes farther removed from the world of sensation" (Weinberg 1976: 28). For instance:

> You know that if your computer beats you at chess, it is really the *program* that has beaten you, not the silicon atoms or the computer as such. The abstract program is instantiated physically as a high-level behaviour of vast numbers of atoms, but the *explanation* of why it has beaten you cannot be expressed without also referring to the program in its own right. That program has also been instantiated, unchanged, in a long chain of different physical substrates, including neurons in the brains of the programmers and radio waves when you downloaded the program via wireless networking, and finally as states of long- and short-term memory banks in your computer. The specifics of that chain of instantiations may be relevant to explaining how the program reached you, but it is irrelevant to why it beat you: there, the content of the knowledge (in it, and in you) is the whole story. That story is an explanation that refers ineluctably to abstractions; and therefore those abstractions exist, and really do affect physical objects in the way required by the explanation.
>
> (Deutsch 2011: 114–115, emphases original)[14]

Consistent with this reasoning, it is not unreasonable to "give a higher degree of reality" to an "abstract mathematical model" of linguistic computation than to its implementation in the "ordinary world of sensation." But this poses a problem for the "Mind, Mechanism and Mathematics" theme of the Turing Centenary:

> The joint study of brain and language […] has achieved some basic results correlating linguistic phenomena with brain responses, but has not advanced to any explanatory theory that identifies the nature of linguistic computation in the brain […]. The absence of an explanatory theory of this type is the result of the conceptual granularity mismatch and the ontological incommensurability between the foundational concepts of linguistics and

[13] This can be construed as a restatement of the Platonic theory of forms.

[14] This is to restate the Aristotelean distinction of matter and form.

> those of neurobiology […]. Consequently, there is an absence of reasonable linking hypotheses by which one can explore how brain mechanisms form the basis for linguistic computation. (Poeppel & Embick 2005: 14–15)

The 'conceptual granularity' in theories of I-languages is measured on 'higher' computational and algorithmic levels whereas the primitives of neuroscience are posited on the 'lower' implementational level (see Marr 1982). *Prima facie*, the ontologies of these levels are incommensurable: set-formation, phrases, and so on in linguistics; action potentials, neurons, and so forth in neuroscience. If a mathematical model of intelligent thought is to be formulated — and *a fortiori* realized as artificial intelligence — a novel and nontrivial unification of the levels of analysis, the levels "at which any machine carrying out an information-processing task must be understood" (Marr 1982: 25), is imperative, requiring interdisciplinary research. The concept that unifies the research is *computation* — essential to which is *information* — and the concept that unifies computation is the Turing machine. Indeed, the beauty of the Turing machine is that in its abstractness it subsumes and thereby relates all computational primitives; in principle therefore it renders commensurable the computational ontologies of linguistics and neuroscience — or so I would endeavor to prove in the TPLT.

A Turing machine is a mathematical abstraction, not a physical device, but my theory is that the information it specifies in the form of I-language must be encoded in the human genetic program — and/or derived from the mathematical laws of nature ('third factors' in the sense of Chomsky 2005) — and expressed in the brain. Central to the machine is a generative procedure for d-infinity; however, "[a]lthough the characterizations of what might be the most basic linguistic operations must be considered one of the deepest and most pressing in experimental language research, we know virtually nothing about the neuronal implementation of the putative primitives of linguistic computation" (Poeppel & Omaki 2008: 246). So is presented the great challenge for the TPLT: To precisify (formalize) the definitions of linguistic primitives in order that 'linking hypotheses' (not mere correlations) to as yet undiscovered neurobiological primitives can be formed.

4. Generative Systems

It was in the theory of computability and its equivalent formalisms that the infinite generative capacity of a finite system was formalized and abstracted and thereby made available to theories of natural language (see Chomsky 1955 for a discussion of the intellectual zeitgeist and the influence of mathematical logic, computability theory, etc. at the time generative linguistics emerged in the 1950s). In particular, a *generative grammar*[15] was defined as a set of rules that recursively generate (enumerate/specify) the sentences of a language in the form of a *production system* as defined by Post (1944) and exapted by Chomsky (1951):

(1) $\phi_1, \ldots, \phi_n \rightarrow \phi_{n+1}$

[15] Linguists use the term with systematic ambiguity to refer to the explananda of linguistic theory (i.e. I-languages) and to the explanantia (i.e. theories of I-languages).

"[E]ach of the ϕ_i is a structure of some sort and […] the relation → is to be interpreted as expressing the fact that if our process of recursive specification generates the structures ϕ_1, …, ϕ_n then it also generates the structure ϕ_{n+1}" (Chomsky & Miller 1963: 284); the inductive (recursive) definition derives infinite sets of structures. The objective of this formalization was analogous to "[t]he objective of formalizing a mathematical theory a la Hilbert, [i.e.] to remove all uncertainty about what constitutes a proof in the theory, […] to establish an algorithm for the notion of proof" (Kleene 1981: 47) (see Davis 2012 on Hilbert's program). Chomsky (1956: 117) observed that a derivation as in (1) is analogous to a proof with ϕ_1, …, ϕ_n as the set of axioms, the rewrite rule (production) → as the rule of inference, and the derived structure ϕ_{n+1} as the lemma/theorem. For a toy model, let (2) be a simplified phrase structure grammar with S = Start symbol *Sentence*, $\frown$ = concatenation, # = boundary symbol, $N[P]$ = Noun [Phrase], $V[P]$ = Verb [Phrase]:

(2) $\#\frown S\frown\#$
$S \rightarrow NP\frown VP$
$VP \rightarrow V\frown NP$
$NP \rightarrow the\frown man, the\frown book$
$V \rightarrow took$

(3) is one possible derivation given the grammar (production system) in (2):

(3) $\#\frown S\frown\#$
$\#\frown NP\frown VP\frown\#$
$\#\frown the\frown man\frown VP\frown\#$
$\#\frown the\frown man\frown V\frown NP\frown\#$
$\#\frown the\frown man\frown took\frown NP\frown\#$
$\#\frown the\frown man\frown took\frown the\frown book\frown\#$

The derivation proceeds deterministically, stepwise, "remov[ing] all uncertainty about what constitutes a [derivation] in the theory".

Restricted and unrestricted production systems were proved by Post (1944, 1947) to be formally equivalent to the effectively calculable functions — and by extension the λ-calculus and by extension Herbrand-Gödel general recursion (Gödel 1934, Church 1936, Kleene 1936) — proved by Turing (1936) to be equivalent to the computable functions.[16,17] These equivalences necessitated additional restrictions on generative grammars (or seemed to): "[A]n arbitrary Turing machine, or an unrestricted rewriting system, is too unstructured to serve as a grammar […]. Obviously, a computer program that succeeded in generating

[16] Independent of Turing, Post (1936) formulated a computational (mathematical) machine equivalent to Turing machines and Gödel-Church recursiveness — and inferred from it provocative psychological conclusions (contradicting those of Penrose 1989).

[17] A Turing machine M can be described by a set Σ of rewrite rules that convert a string $\#S_0\phi\#$ to $\#S\#$ just in case M accepts $\#\phi\#$. (Given the determinism of M, Σ is monogenic.) The set of rewrite rules Σ' containing $\psi \rightarrow \chi$ just in case $\chi \rightarrow \psi$ is in Σ and containing a Stop rule $\#S_0 \rightarrow \#$ is an unrestricted rewriting system.

sentences of a language would be, in itself, of no scientific interest unless it also shed some light on the kinds of structural features that distinguish languages from arbitrary, recursively enumerable sets" (Chomsky 1963: 359–360).[18]

Thus the Chomsky hierarchy of grammar types — type 0 (unrestricted ≈ Turing machines) ⊃ type 1 (context sensitive ≈ linear bounded automata) ⊃ type 2 (context free ≈ pushdown automata) ⊃ type 3 (finite state ≈ finite automata) — was formulated to define the generative capacities of grammars and corresponding automata (see Chomsky 1956). The objective for theoretical linguistics was then — *and should be again* — to discover the type of grammar/automaton for natural language; language defined as a cognitive system realizing a Turing machine subtype. But this research program failed because of a preoccupation with weakly generated strings (sentences) rather than strongly generated structures. In the original conception of the hierarchy (see Chomsky 1959a), but not since (see Boden 2006), it was understood that merely enumerating sentences was of no interest to the empirical science of natural language: "Along with a specification of the class *F* of grammars, a theory of language must indicate how, in general, relevant *structural information* can be obtained for a particular sentence generated by a particular grammar" (Chomsky 1959a: 138, emphasis added). Such a theory of the hierarchy has *yet* to be formulated. A novel mathematical model — a hierarchy revamped for strong generation — is necessary; hence the proposed TPLT.

With such mathematical formalism, I would not be "play[ing] mathematical games", but rather "describ[ing] reality" (Chomsky 1955: 81), which is mathematical: That is, I would not be precisifying mathematically a theory of a non-mathematical system, but formulating a mathematically precise theory of a cognitive system that is mathematical; the theory needs to be mathematical because the phenomenon is mathematical (see Turing 1954). With such apparatus, generative grammar substantializes the romantic intuition of language as "the infinite use of finite means" (von Humboldt 1836: 122).

Intuition is to be *explained* by a theory of linguistic computation; it is to be derived (as an effect), not stipulated (as a cause).[19] This "move from the intuitive hints and examples of traditional grammar to explicit generative procedures" (Chomsky 1995: 24) was but a subroutine in the general research program starting with the demonstration that "the computable numbers include all numbers which could *naturally* be regarded as computable" (Turing 1936: 230, emphasis added). Indeed, in constructing abstract machines, Turing and Chomsky — like all theorists of algorithms — were exorcising ghosts: "If the

---

[18] See Pinker (1979) for a discussion on the (un)decidability and (un)learnability of recursively enumerable sets.

[19] Explanations must 'go all the way down', a nontrivial truism: "In imagining that there is a machine whose construction would enable it to think, to sense, and to have perception, one could conceive it enlarged while retaining the same proportions, so that one could enter into it, just like into a windmill. Supposing this, one should, when visiting within it, find only parts pushing one another, never anything by which to explain a perception" (Leibniz 1714: 83). If intuition were 'in control' of the cognitive-neurobiological windmill, an explanation for *it* would be necessary, so the explanation must go all the way down, but where it 'bottoms out' is a *mystery* — perhaps one of "[nature's] ultimate secrets [stored in] that obscurity, in which they ever did and ever will remain" (Hume 1763: 323).

grammar is […] perfectly explicit — in other words, if it does not rely on the intelligence of the understanding reader but rather provides an explicit analysis of his contribution — we may […] call it a *generative grammar*" (Chomsky 1965: 4); "[w]e may compare a man in the process of computing a real number to a machine which is only capable of a finite number of [configurations]. If at each stage the motion of the machine […] is *completely* determined by the configuration, we shall call the machine an 'automatic machine'" (Turing 1936: 231–232, emphasis original), now called a Turing machine.

With the ghost exorcised, the elementary components and procedures of generative grammar must reduce to the elementary components and procedures of the Turing machine, consistent with the thesis that if a function f is effectively calculable, then f is computable by a Turing machine, hence by the Universal Turing machine — the computer which can assume the form of any Turing machine M given the mathematical-functional description (machine table) of M as input.[20] The formulation of a generative grammar as a Turing machine would be novel to the TPLT and necessary for progress in discovering — and realizing artificially — the system for cognitive-biological computation.

Generative grammar was initially formulated as a Post-production system, but that project in "mathematical linguistics is [now] in a plateau", having "got about as far as it could from the point of view of any impact on [empirical] linguistics" (Chomsky 2004b: 68–69) because now "virtually the entire subject [of mathematical linguistics] deals with weak generation; strong generation, while definable, is [now] too complex for much in the way of mathematical inquiry" (Chomsky 2007: 1097). Only strong generation encodes grammatical information, and therefore mathematical linguistics must be formulated in its terms to impact empirical linguistics.

One problem is that a formulation of strong generative capacity in Post-production systems is infeasible. However there is a different and deeper problem: Even strong generation is insufficient for empirical adequacy. As with weak generation, a grammar defined only in terms of strong generative capacity is a function defined in extension (i.e. defined in terms of expressions generated) when the desideratum of linguistic theory is a function defined in intension (i.e. defined as the generator of expressions). The reasons for this desideratum are self-evident. First, the enumeration of the extensional set is possible only by running the procedure for the intensional function; thus the logical priority of the latter over the former.[21] Second, the ontological status of the sets of expressions is debated (see Chomsky 1986, 2001, Postal 2004, 2009), but the reality of the generator is manifest in the behavior of any normal human engaged in linguistic creativity, the 'infinite use of finite means'. It is this function in intension that evolved in the brain; the complexity of Post-production systems cannot be posited with plausibility in theories of neurobiology and its evolution (see Ber-

[20] Turing discovered that a program can be a form of data. The profundity of this equation has not yet been appreciated in biolinguistics.

[21] Gödel recognized the extensional equivalence of Herbrand-Gödel recursiveness, λ-calculus, and Turing machines, but by intensional analysis was convinced that, in his words, "the correct definition of mechanical computability was established beyond any doubt by Turing" (see Soare 2009).

wick 2011).[22]

A novel theory of generative capacity is necessary, and here "Turing's computability is intrinsically persuasive in the sense that the ideas embodied in it directly support the thesis that the functions encompassed are all for which there are algorithms" (Kleene 1981: 49). If the project is to unify the computational, algorithmic, and implementational levels of language — and ultimately intelligence — then the Turing machine is ideal, for it defines the mathematical-functional components and procedures that any computational system *must* realize: "[Turing's] concepts underlying the design of computing machines arise out of a kind of *conceptual necessity*. [I]f one analyzes any computing machine that is powerful, fast, and efficient, one *will* find these concepts realized in its functional structure. [And yet Turing's concepts] have been largely ignored in contemporary efforts to imagine how the brain might carry out the computations that the behavioral data imply it does carry out" (Gallistel & King 2009: 125, 144, emphases added).[23]

Linguistic computation is demonstrably powerful, fast, and efficient, and yet contemporary biolinguistics — indeed all of theoretical linguistics and neurolinguistics — has largely ignored Turing's concepts. If I-language were precisified as a form of Turing machine, our imagination for how I-language computes in mathematical abstraction and how it relates to concrete computation in a brain or machine would be profoundly expanded — so as, perhaps, to compass the truth. Such an expansion of the imagination would be the goal of the TPLT.

> [Imagination] is that which penetrates into the unseen worlds around us, the worlds of Science […]. Those who have learned to walk on the threshold of the unknown worlds […] may then with the fair white wings of Imagination hope to soar further into the unexplored amidst which we live.
>
> (Lady Lovelace, Babbage's programmer, in Gleick 2011)

Research at the abstract computational level constrains research at the concrete implementational level: For instance, if research on the former level demonstrates that the mind *does* generate d-infinity, then it is incumbent upon research at the latter level of the brain to demonstrate *how*.[24,25] Failure to concede

---

[22] Some neurobiologists (e.g., Zylberberg *et al*. 2011), do posit 'production rules' in neuronal computation, but these are not Post-productions.

[23] I cannot overemphasize the fact that the Turing machine represents the *multiply realizable mathematical-functional structure* of a computer: "Since Babbage's machine was not electrical, and since all digital computers are in a sense equivalent, we see that this use of electricity [in brains and computers] cannot be of theoretical importance […]. If we wish to find […] similarities we should look rather for *mathematical analogies of function*" (Turing 1950: 439, emphasis added); "[t]he brain is a purely physical object made up of completely sterile and inanimate components, all of which obey exactly the same laws as those that govern the rest of the universe. The key is not the *stuff* out of which brains are made, but the *patterns* that can come to exist inside the stuff of the brain. Brains are *media* that support complex patterns" (Hofstadter 1979: P-4, emphases original).

[24] Analogous logic has proved successful elsewhere in science and needs to be assumed in the sciences of mind and brain: "[Computational-Representational] theories [of the mind] will provide guidelines for the search for [neurophysiological] mechanisms, much as nineteenth-century chemistry provided crucial empirical conditions for radical revision of fundamental physics. The common slogan that 'the mental is the neurophysiological at a higher level' —

this top-down logic has retarded research, as with "[c]onnectionists draw[ing] their computational conclusions from architectural commitments", which is fallacious given our limited understanding of the mathematical-functional meaning of neuronal architecture, "whereas computationalists draw their architectural conclusions from their computational commitments" (Gallistel & King 2009: ix), which is sound because of our greater understanding of computation, due to Turing (see Vaux & Watumull 2012 for a critique of connectionism as implemented in Optimality Theory phonology). The understanding of mental software thus precedes and conditions the understanding of neurophysiological hardware: "[I]t is the mentalistic studies that will ultimately be of greatest value for the investigation of neurophysiological mechanisms, since they alone are concerned with determining abstractly the properties that such mechanisms must exhibit and the functions they must perform" (Chomsky 1965: 193).

Hence it is necessary, as neurobiologists Zylberberg *et al*. (2011: 294) argue, for research at the lower level to "investigate which neural architecture could implement a Brain Turing Machine", which neural architecture could implement "the concept of a production […] essentially equivalent to the action performed by a Turing machine in a single step", necessitating a "selection of productions […] determined by the contents of working memory, which plays the role of the tape in the Turing machine […]. Iteration of the cycle of production selection and action constitutes the basis of Turing-like programs".[26] For the 'Brain Turing Machine' research program to succeed, the objects of inquiry must be identified by novel and precise higher level definitions of Turing's components and procedures (e.g., 'tape', 'production', etc.). To formulate such definitions, defining objectives to unify the sciences of mind and brain would be my ambition for the TPLT. As of yet, I can proffer merely quasi-formal definitions and simplified (simplistic) prototypes of theories and models that in future research could answer big questions of minds, machines, and mathematics.

---

where C-R theories are placed within the 'mental' — has matters backwards. It should be rephrased as the speculation that the neurophysiological may turn out to be 'the mental at a lower level'" (Chomsky 2000: 25–26).

[25] Given the evidence for cognitive computation, it is in my judgment logically necessary to assume the language faculty to be a form of Turing machine with addressable read/write memory. This logic applies to the brain generally (Gallistel & King 2009: 125, 105, i): If "the brain is an organ of computation[,] then to understand the brain one must understand computation", which necessitates formalization; "[Turing] created a formalization that defined a class of machines", with mathematical-functional components and procedures so elementary as to be multiply physically realizable. And "[b]y mathematically specifying the nature of these machines, and demonstrating their far-reaching capabilities, [Turing] laid a rigorous foundation for our understanding of what it means to say something is computable". A formalization can in this way define conditions of adequacy that any theory in a particular domain of inquiry must satisfy to be true. Therefore, if it is demonstrated by research on higher levels of analysis that "brains are powerful organs of computation", and that a formally definable "[addressable read/write] memory mechanism is indispensable in powerful computing devices", then we must demand of neuroscience an implementational theory of an addressable read/write memory.

[26] Interesting and unorthodox work in neurobiology (e.g., Gallistel & King 2009, Hameroff & Penrose 1996) speculates that subcellular systems (e.g., in microtubules) could be efficiently implementing components and procedures of the Brain Turing Machine.

5. The Linguistic Turing Machine

A Turing machine is not a physical device — nor an intended model/simulation thereof — but rather a mathematical abstraction representing the functional conditions necessary and sufficient for any system — including the brain — to be computational. Let the components of the linguistic Turing machine *L* be a control unit, a read/write head, and a tape.

(4) *The linguistic Turing machine L is the 5-tuple* $(Q, \Gamma, \delta, \#_S, \#_H)$[27,28]

Q: Set of states/instructions (i.e., principles and parameters)
Γ: Set of symbols for syntactic objects (e.g., lexical items, phrases, etc.)
δ: $Q \times \Gamma \rightarrow Q \times \Gamma$ (i.e. transition function from state/symbol to state/symbol by search/merge)
$\#_S$ ($\in \Gamma$): Start (boundary) symbol
$\#_H$ ($\in \Gamma$): Halt (boundary) symbol

The potentially infinite bidirectional tape represents the inputs and outputs (memory) of *L and* stores its program (which can thereby be modified in language acquisition); for empirical adequacy (e.g., syntactic transformations), the symbolic memory of the tape can be structured as a stack with (a restricted form of) random access (NB: *L* is not a classical Turing machine; it is domain-specific and could be equipped with multiple tapes/stacks, etc.). The control unit is a transition function (machine table) defining a finite set of states and a finite set of instructions for the operation of *L* given a state and an input represented on the tape and/or possibly a symbol on the stack memory; the states and instructions are defined by universal linguistic principles and parameterization for particular natural languages (see Roberts 2011 for a computational theory of parametric variation with principled constraints). The read/write head is a search/merge procedure: Effectively, a rewrite rule of the form in (1); it reads (searches for) an input symbol (or symbols) on the tape/stack and, as commanded by the control unit, writes (merges) an output symbol on the tape (equivalently, pushes a

[27] Different but equivalent definitions could be formulated. A formulation of the search/merge procedure in Polish notation could precisify the efficiency of its execution (Randy Gallistel, p.c.).

[28] States for acceptance and rejection are not specified here, *tentatively*, because the acceptance (as grammatical) and rejection (as ungrammatical) of expressions are, *arguably*, functions of interpretability conditions at the interfaces with the conceptual-intentional and sensory-motor systems; the generation and interpretation of an expression are, *arguably*, independent processes in principle (but in practice the latter can condition the former). And at the interfaces, an approximately continuous scale of deviance — not discrete states of grammaticality (acceptance/rejection) — would probably be required: Some technically grammatical sentences are intuitively rejectionable and some intuitively acceptable sentences are technically ungrammatical and some sentences are, on some passes, of indeterminate status. As Putnam (1961) argues, and I concur, grammaticality judgments could be formally equivalent to decidability problems; perhaps, therefore, some judgments are technically undecidable, but this must be investigated in the TPLT. In addition to these complexities are Kripke's (1982) 'Wittgensteinian problems' (see Chomsky 1986), such as the issue of ungrammaticalities being 'mistakes', cases of not 'following the rules' of grammaticality; we can wonder whether a linguistic Turing machine could even 'make a mistake' (see Turing 1950). These would be interesting problems to address in the TPLT.

symbol onto the stack) in a stepwise process; and the rules themselves, stored as a program, can be written-to/read-from — a fact with profound implications for theories of the evolution of I-language, language change, language growth/acquisition, syntactic computations, *inter alia*, to be investigated in the TPLT.

To recapitulate, a Turing machine is a mathematical abstraction, not a physical device, but my theory is that the information it specifies in the form of I-language (4) must be encoded in the human genetic program — and/or given by mathematical law — and expressed in the brain. Central to the machine is the search/merge procedure, and yet "[a]lthough the characterizations of what might be the most basic linguistic operations must be considered one of the deepest and most pressing in experimental language research, we know virtually nothing about the neuronal implementation of the putative primitives of linguistic computation" (Poeppel & Omaki 2008: 246). Thus is presented the great challenge for the TPLT: To precisify (formalize) the definitions of linguistic primitives so that 'linking hypotheses' (not mere correlations) to neurobiological primitives — constituative of a Brain Turing Machine — can be formed. The beauty of the Turing machine is that in its abstractness it subsumes and thereby relates all computational primitives; in this way it could render commensurable the computational ontologies of linguistics and neuroscience. And the key to linking these ontologies is the generative procedure.

Rewrite rules of the form in (1) were defined in Chomsky normal form (5); this can be updated for *L*, with the search/merge procedure — call it *Merge* — formulated as in (6).

(5) Chomsky normal form: Uppercases represent nonterminals (e.g., *S*, *VP*, *NP*, etc.), lowercases represent terminals (e.g., *the*, *man*, *took*, etc.), ϵ represents the empty string.

$$A \rightarrow BC$$
$$A \rightarrow a$$
$$S \rightarrow \epsilon$$

(6) Merge is a set-formation function: The syntactic objects α, β, γ can be simple (e.g., lexical items) or complex (e.g., phrases) such that the nonterminal/terminal distinction is unformulable; consequently, the system is simplified and generalized.[29,30]

$$\#_S \rightarrow \{\alpha, \#_S\}$$
$$\{\alpha, \#_S\} \rightarrow \{\beta, \{\alpha, \#_S\}\}$$
$$\{\gamma, \#_S\} \rightarrow \{\#_H, \{\gamma, \#_S\}\}$$

[29] As a component of a Turing machine, the Merge procedure is necessary and sufficient to implement the elementary operations of arithmetic (consistent with the composition of functions and set-theoretic definitions of the natural numbers); and interestingly, with Merge defined as a binary operator on the set of syntactic objects, *L* can be formulated as a free magma in universal algebra.

[30] Boundary (Start/Halt) symbols are logically necessary for any computational system and can be demonstrated (see Watumull 2012) to solve problems with syntactic structures (e.g., labeling and linearizing the first merged elements, transferring/spelling-out the final phase, structuring inter-sentential coordination).

To simplify: Merge(X, Y) = {X, Y}.[31] A simplified derivation by Merge of *the man took the book* in (3) assumes the form in (7): The predicate and subject are generated by Merge in parallel and then merged:

(7)

$\#_S$	$\#_S$
{*book*, $\#_S$}	{*man*, $\#_S$}
{*the*, {*book*, $\#_S$}}	{*the*, {*man*, $\#_S$}}
{*took*, {*the*, {*book*, $\#_S$}}}	{$\#_H$, {*the*, {*man*, $\#_S$}}}

{{$\#_H$, {*the*, {*man*, $\#_S$}}}, {*took*, {*the*, {*book*, $\#_S$}}}}
{$\#_H$, {{$\#_H$, {*the*, {*man*, $\#_S$}}}, {*took*, {*the*, {*book*, $\#_S$}}}}}

It is obvious from (7) that strongly generative — structure (set) forming — recursion is necessary to generate any nontrivial expression: Information from step *i* must be carried forward for merger at step *i* + 1; as a Turing machine, *L* "carries symbolized information forward […], making it accessible to computational operations" (Gallistel & King 2009: 122).[32] A finite state automaton, which "cannot write to tape [or push to a stack] cannot store the results of its computations in memory for use in subsequent computations" (Gallistel & King 2009: 122), is provably inadequate (see Chomsky 1956). Nor is a connectionist network possible (probably), because a look-up table architecture — to which connectionist networks reduce (arguably) — cannot (or can only inefficiently) implement numerous linguistic phenomena such as d-infinity and phonological rules (see Vaux & Watumull 2012).

Given the generality of Merge, it is not obvious that I-language is *not* as powerful as an unrestricted Turing machine. Nor is it obvious that I-language *is* computable in general. The implications of incomputability would obviously be profound for our understanding not only of language, but of nature generally (see Cooper 2012) — implications to be explored in the TPLT. It is intriguing though that in the history of generative linguistics, solutions to empirical problems reducible to but usually unrecognized as incomputability problems unfailingly but unnoticeably reestablished the computability of I-language.

---

[31] Merge entails 'external' and 'internal' forms: $EM(X, Y \mid X \notin Y \wedge Y \notin X) = \{X, Y\}$; $IM(X, Y \mid X \in Y \veebar Y \in X) = \{X, Y\}$. EM corresponds to phrase structure rules (and argument structure semantics), IM to transformation/movement rules (and discourse semantics).

[32] Recursively generated hierarchy can be super-sentential (i.e. Merge can structure discourse representations) and realized in pushdown automata:

> In conversational storytelling, […] embedded stories are normally flashbacks or flashaheads. The system [in state *q*] recognizes flash markers on the incoming clause, normally pluperfect marking for flashbacks (accompanied perhaps by a temporal deictic comment like 'before this…') and present progressive or future tense plus comments for flashaheads. The flash marker effects a PUSH to the embedded story [state *q'*] which parses subsequent clauses as [stative] or [eventive] and stores them in […] registers. When a clause arrives for parsing which indicates the end of the flash with a 'so,' 'anyway,' 'and then,' [etc.], which act as POP markers, the parse of the embedded unit is complete and the parser returns to [*q*] to accept further input.
>
> (Polanyi & Scha 1983: 164–165)

The recursive generation of discourse would be a research priority in the TPLT as it has not been in generative linguistics hitherto.

Indeed, I would argue — and expound in the TPLT — that the computability of I-language would be strong evidence against 'Super-Turing machines' and 'hyper-computation' (in I-language and beyond); I would concur that the existence of non-Turing processes is a "myth" (Davis 2004). So I assume that with additional work, I could prove Merge (as formulated in (6)) to be computable and tractable (i.e. efficient in being upper-bounded by a polynomial function) and optimal.[33]

The emergence of optimality — even beauty — fascinated Turing (1952); he thus would have been interested in the fact that *binary* Merge is optimal in minimizing formal representations and spatiotemporal resources — measurable by Kolmogorov complexity and similar concepts — in the process of maximizing the strong generation of syntactic structures, consistent with the minimax theorem (see von Neumann 1928, Watumull 2010).[34,35]

It has been demonstrated mathematically (see Turing 1952 on morphogenesis, Mandelbrot 1982 on fractals, Chaitin 2012 on natural selection) and simulated computationally (see Minsky 1985 on Turing machines, Wolfram 2002 on cellular automata) that evolution does tend to converge on simple procedures capable of generating infinite complexity. These demonstrations and simulations need to be applied in biolinguistics and unified in explanation; this would be a project in the TPLT.

Watumull *et al*. (in preparation) conduct a rigorous mathematical and biological analysis of the hypothesis that a "computational mechanism of recursion", such as Merge, "is recently evolved and unique to our species" and unique to language (Hauser *et al*. 2002: 1573).[36] It has been assumed, dubiously, that this controversial hypothesis can be adjudicated empirically with counter-evidence that some language does not run recursive computation (see Everett 2005) or that linguistic universals do not exist at all (see Evans & Levinson 2009). This assumption is false (as Watumull *et al*. explain), but granting it *arguendo*, my formalization of I-language as the linguistic Turing machine *L entails* the properties of the humanly unique "computational mechanism of recursion" as one of many linguistic universals; this is satisfying because "[t]he most interesting contribution a generative grammar can make to the search for universals of language is specify formal systems that have putative universals as *consequences*, as opposed to merely providing a technical vocabulary in terms of which autonomously stipulated universals can be expressed" (Gazdar *et al*. 1985: 2, emphasis original).

As formulated in (6), Merge is necessarily recursive, automatically generative of hierarchically structured expressions over which set-theoretic relations relevant to linguistic cognition (e.g., c-command, probe–goal) are defined, not stipulated. These properties derive as logical consequences from the formulation of Merge as the simplest read/write mechanism — logically necessary in any pow-

---

[33] See Watumull (2012) for examples of (in)computable linguistic phenomena, arguments against Super-Turing machines, and the optimality/efficiency of syntactic computations.

[34] Syntactic structures are even fractal (see Watumull & Eglash 2011).

[35] I argue (Watumull 2010) that Merge is a *minimax function*: Effectively unary (compact) but binary (combinatorial).

[36] In brief, the sense of *recursion* in Hauser *et al*. (2002) is that of an effective procedure fixing a computable function that enumerates and maps to interfacing systems an infinite set of possible expressions.

erful/productive and economical/efficient formal system: Recursively generated hierarchy defined by structural (grammatical) relations is *deduced* — not merely predicted — to be a linguistic universal. To formalize this informal deduction, deriving additional linguistic universals, would be at the core of the TPLT.[37,38]

6. Acquisition

One of the biggest questions for the TPLT is the derivation of knowledge, a question embedded in the even bigger question "whether there is a characteristic human psychological phenotype ('human nature' in earlier editions) that can be attributed to a characteristic human genetic endowment" (Fodor 2001: 102). If the thesis of the TPLT that I-language is a type of Turing machine is correct, then it is evident that the elementary components and procedures of the linguistic Turing machine *L* must be preinstalled in — genetically specified for — the brain and/or given by mathematical law, and thus characterisitic of humans. Equally evident is the fact that variation exists in its implementation and execution: I-language is a species-typical property of *individuals*, so technically there exist as many languages as there exist (and have existed) individuals, and if a sufficient number of individuals converge (approximately) on a specific implementation and execution of *L*, they are classified as knowing the 'same' language (e.g., 'English', 'Swahili', etc.). This variation is a function of the fact that some parts of I-languages must be acquired ('learned') in an astonishing process that is not yet understood by science:

> A normal child acquires [linguistic] knowledge on relatively slight exposure and without specific training. He can then quite effortlessly make use of an intricate structure of specific rules and guiding principles to convey his thoughts and feelings to others, arousing in them novel ideas and subtle perceptions and judgments. For the conscious mind [of the scientist], not specially designed for the purpose, it remains a distant goal to reconstruct and comprehend what the child has done intuitively and with minimal effort. Thus language is a mirror of mind in a deep and significant sense. It is a product of human intelligence, created anew in each individual by operations that lie far beyond the reach of will or consciousness.
>
> (Chomsky 1975: 4)

This is but a particular articulation of the general question posed by Bertrand Russell (1948: v, emphasis original): "*How comes it that human beings, whose contacts with the world are brief and personal and limited, are nevertheless able to know as*

---

[37] A profound question in computational linguistics and computational neuroscience is whether linguistic rules are structure-dependent (defined over sets) or linear-dependent (defined over strings) and why (see Perfors *et al*. 2011, Berwick *et al*. 2011). My answer (i.e. rules are structure-dependent because of the set-theoretic relations entailed by the running of Merge), can be deduced if the Turing machine *L* does in fact represent linguistic computation. I would show this in the TPLT.

[38] An additional property of language predicted — deduced — to be universal on my model is that of binarity: Merge as defined for *L* is a binary function such that the expressions it generates are composed of (embedded) 2-sets. I can demonstrate that Merge of any arity n, $n > 2$, is either incomputable or intractable (lower bounded by an exponential function) (see Watumull 2012). This is interesting because many natural languages are argued to contain non-binary structures.

much as they do know?". Any theory of mind and machine needs to answer this question. And by dint of logic — let alone empirical observation — the answer needs to be *specific* to the cognitive domain:

> From a computational point of view, the notion of a *general purpose* learning process (for example, associative learning), makes no more sense than the notion of a general purpose sensing organ. [P]icking up information from different kinds of stimuli — light, sound, mechanical, and so on — requires organs with structures shaped by the specific properties of the stimuli they possess […]. Learning different things about the world from different kinds of experience requires computations tailored both to what is to be learned and to the kind of experience from which it is to be learned […]. No one would suppose that [a path integration learning organ] would be of any use in the learning of a language. Applying this learning organ to that learning problem would be like trying to hear with an eye or breath with the liver.
>
> (Gallistel 2010: 194–196, emphasis added)

Language acquisition is patently a computational process necessitating procedures for the recognition and analysis of linguistic data — out of the "blooming, buzzing confusion" (James 1890: 488) of sensory stimuli — so as to map the information into the linguistic Turing machine, which must, within a "critical period" (see Lenneberg 1967, Pinker 1984, 1994), parameterize the initial state of its control unit — interpolating language-particular instructions between (genetically-installed/mathematically-given) language-universal principles — so as to generate outputs consistent with the inputs and ultimately attain (grow) a steady state of linguistic competence. To discover the mechanisms of this process, its complexity can be abstracted into the simplicity of a Turing machine, but as of yet, the abstraction has not been conducted — another job for the TPLT.

Reinforcing the classic argument from the poverty of the stimulus (see Chomsky 1980) — that is, an innate language acquisition device LAD is necessary because the linguistic competence the child displays is underdetermined by the linguistic data to which it is exposed — is an argument from the complexity of the computation (see Aaronson 2011). For instance, let *C* be a child of generation *g* unequipped with the domain-specific LAD and exposed to a set of *n*-bit strings (sentences) weakly generated by paths in a nondeterministic finite automaton *M* (the grammar of generation *g*-1) with a number of states < *n*.[39] It has been proved mathematically (see Kearns & Valiant 1994) that if *C* can reconstruct *M* — if the child can form the grammar — then *C* can break the RSA cryptosystem, which is technically intractable.[40] "The grammar of any real human language is much too rich and complicated to be captured by a finite automaton. So this result is saying that even learning the least expressive, unrealistically simple language is already as hard as breaking RSA" (Aaronson 2011). Neither *C* nor any efficient algorithm

---

[39] The primary linguistic data to which the child is exposed are strings from which the strongly generated structures must be reconstructed. But this reconstruction is technically an 'ill-posed problem' if no LAD is assumed — indeed incomputable in general as stated in Rice's Theorem (see Batchelder & Wexler 1979 for a related discussion).

[40] RSA encodes messages on the Internet by raising the encoded message to an exponent modulo a composite (semiprime) number *p* with neither of its prime factors given. To decode the message, it is necessary (though not sufficient) to factor *p* into two primes.

yet formulated by computer scientists can break RSA,[41] but *C* (assuming it to be a normal child) attains/grows an I-language. It follows that *C* must be genetically preprogrammed with some type of LAD.[42]

The LAD could run the 'semantic bootstrapping' algorithm rigorously and empirically defined and expounded by Steven Pinker (1984). The LAD (genetically) en-codes a set of semantic primitives that recognize syntactic categories (genetically specified) given in the linguistic input. With the categories recognized, the child can infer some of the rules particular to the language expressed in the input; the child can then apply these rules to additional input, recognizing additional categories and acquiring additional rules, etc., so as ultimately to attain (grow) the steady state. This bootstrapping logic was formalized (Pinker 1984), but needs to be updated into a Turing machine to prove its success in satisfying conditions any theory of the LAD must satisfy (Pinker 1979): (i) *learnability* (languages are acquired); (ii) *equipotentiality* (the LAD must run for all natural languages); (iii) *time* (languages are acquired in efficient time); (iv) *input* (the input information in the theory must be realistic); (v) *development* (the stages of acquisition are predict-able); (vi) *cognitive* (the theory of the LAD must be compatible with theories of extralinguistic cognitive faculties). These conditions could be efficiently satisfied in a PAC algorithm (see Valiant 1984) running in a hierarchy of linguistic parameters (see Roberts 2011).

A PAC ("probably approximately correct") model is an efficient, polynomial (not exponential) bounded algorithm presented with a set of (random) points $x_1, \ldots, x_n$ from some set S with the points classified $f(x_1), \ldots, f(x_n)$. The model succeeds if the function f is inferred such that $f(x)$ can be predicted for the majority of future points $x \in S$. The function f is a member of a hypothesis class H (the set of possible hypotheses). The form of H is important:

> [T]he PAC approach has the advantage [over a Bayesian approach] of requiring only a *qualitative* decision about which hypotheses one wants to consider, rather than a *quantitive* prior over hypotheses.
>
> (Aaronson 2011, emphases original)

For language (see de Wolf 1997), let the domain S be the set of all possible sentences (which in the TPLT I would define in terms of strong generation); a function f is a grammar such that H is the set of possible grammars (genetically specified and mathematically governed); the child needs to succeed in converging on f given a set of sentences $x_1, \ldots, x_n$. The algorithm can *query* an *oracle* device (Turing 1939) as to the membership of a given x in S for a given f; an oracle is a 'black box' connected to a Turing machine to which queries can be entered and from which answers are returned. For language, the child does not query persons in its external environment (obviously), but (unconsciously) of internal representations of syntactic structure; those representations genetically specified constitute the 'oracular information'. ('Universal Grammar' is a theory of the genetically installed linguistic oracle.) Polynomial PAC acquisition of type

---

41 I set aside Shor's quantum algorithm (see Shor 1997).

42 See Pinker (2002) on the problems with presuming the child brain to be equipped with "little mechanism and lots of blank sheets" such that a learning machine can be "easily programmed" (Turing 1950: 456).

3 languages is possible only with membership queries (see Angluin 1987); it has not been proved — *yet* — that type 2 or type 1 languages can be acquired even with membership queries.[43]

A proof could be possible if these grammars are reformulated in the linguistic Turing machine because the structures Merge generates in parsing (i.e. the reverse derivations the child performs in recognizing and analyzing linguistic data) necessitate querying to be interpretable (i.e. mappable into the initial (genetically determined) state of I-language). Some answers are given in the structure (e.g., those defining arrangements of phrases, etc.), obviating oracle queries; residual answers need to be given in the genetic endowment as oracular information (e.g., the semantic and syntactic primitives of Pinker 1984) and/or derived from mathematical law. So designed, acquisition is the process of running a bootstrapping PAC algorithm on a hierarchy of queries, the answers to which specify the setting of language-particular parameters. Ian Roberts has established and is expounding a rigorous research program on parameter hierarchies that derives linguistic universals. Thus, it could be that the "innate [linguistic system] contains little more than the single combinatorial operation Merge and a schema for syntactic categories and features" (Roberts 2011).

In sum, I-language could reduce to the Turing machine *L*: To prove this — and ergo prove the possibility of a unification via mathematics of mind and machine — would be the triumph of the TPLT.

7. The Big Answers

The success of the TPLT, however small, could constitute a proof of concept for a strategy to answer the Big Question of the Turing Centenary as to whether there could be "a successful mathematical model of intelligent thought". I-language is an organ of human intelligence, and I submit that a successful mathematical model of linguistic cognition is within reach.

> [O]ut of a lot of the very interesting work now going in [biolinguistics] there may come a sharpening and clarification of notions which would make it possible to undertake a new mathematical study, and maybe a lot of the questions that were asked about phrase-structure grammars [in the 1950s–1960s] could now be re-asked and many new questions could be asked about these new […] systems. At that point one would hope that there would develop a new upsurge of mathematical linguistics and at that point I think […] mathematical logic might very well be relevant. Ideas from recursive function theory and model theory and the theory of more elaborate logics seem at least to be in the same neighborhood as the kind of formal questions suggested by language study and it might turn out that a new field would develop out of this convergence. (Chomsky 2004b: 69)

To work for such a convergence would be to share and substantiate Turing's dream that not only can we model nature mathematically, but we may even discover that, fundamentally, nature (reality) *is* mathematical.

---

[43] Until the Chomsky hierarchy is updated in the TPLT in terms of strong generation, it is not possible to accept or reject these proofs as conclusive.

References

Aaronson, Scott. 2011. Why philosophers should care about computational complexity. Ms., MIT, Cambridge, MA.

Angluin, Dana. 1987. Learning regular sets from queries and counterexamples. *Information and Computation* 75, 87–106.

Batchelder, William H. & Kenneth Wexler. 1979. Suppes' work in the foundations of psychology. In Radu Bogdan (ed.), *Patrick Suppes*, 149–186. Dordrecht: Reidel.

Berwick, Robert C. 2011. Syntax *facit saltum redux*: Biolinguistics and the leap to syntax. In Anna Maria Di Sciullo & Cedric Boeckx (eds.), *The Biolinguistic Enterprise*, 65–99. Oxford: Oxford University Press.

Berwick, Robert C., Paul Pietroski, Beracah Yankama & Noam Chomsky. 2011. Poverty of the stimulus revisited. *Cognitive Science* 35, 1207–1242.

Bloomfield, Leonard. 1926. A set of postulates for the science of language. *Language* 2, 153–164.

Boden, Margaret. 2006. *Mind as Machine: A History of Cognitive Science*. Oxford: Oxford University Press.

Carnap, Rudolf. 1955. Meaning and synonymy in natural languages. *Philosophical Studies* 6, 33–47.

Chaitin, Gregory. 2012. *Proving Darwin: Making Biology Mathematical*. New York: Pantheon.

Chomsky, Noam. 1951. The morphophonemics of modern Hebrew. Master's thesis, University of Pennsylvania, Philadelphia.

Chomsky, Noam. 1955. The logical structure of linguistic theory. Ms., Harvard University/Massachusetts Institute of Technology, Cambridge, MA.

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions of Information Theory* IT-2: 113–124.

Chomsky, Noam. 1959a. On certain formal properties of grammars. *Information and Control* 2, 137–167.

Chomsky, Noam. 1959b. Review of *Verbal Behavior*, by B.F. Skinner. *Language* 35, 26–57.

Chomsky, Noam. 1963. Formal properties of grammars. In R. Duncan Luce, Robert R. Bush & Eugene Galanter (eds.), *Handbook of Mathematical Psychology II*, 323–418. New York: Wiley and Sons.

Chomsky, Noam. 1964. *Current Issues in Linguistic Theory*. The Hague: Mouton.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Chomsky, Noam. 1975. *Reflections on Language*. New York: Pantheon.

Chomsky, Noam. 1980. *Rules and Representations*. New York: Columbia University Press.

Chomsky, Noam. 1986. *Knowledge of Language: Its Nature, Origins and Use*. New York: Praeger.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Chomsky, Noam. 2000. *New Horizons in the Study of Language and Mind*. Cambridge: Cambridge University Press.

Chomsky, Noam. 2001. Derivation by phase. In Michael Kenstowicz (ed.), *Ken*

Hale: A Life in Language, 1–52. Cambridge, MA: MIT Press.
Chomsky, Noam. 2002. *On Nature and Language*. Cambridge: Cambridge University Press.
Chomsky, Noam. 2004a. Beyond explanatory adequacy. In Adriana Belletti (ed.), *Structures and Beyond: The Cartography of Syntactic Structures,* vol. 3, 104–131. Oxford: Oxford University Press.
Chomsky, Noam. 2004b. *Language and Politics* (ed. by Carlos Otero). Oakland, CA: AK Press.
Chomsy, Noam. 2005. Three factors in language design. *Linguistic Inquiry* 36, 1–22.
Chomsky, Noam. 2007. Review of Margaret Boden's *Mind as Machine: A History of Cognitive Science*. *Artificial Intelligence* 171, 1094–1103.
Chomsky, Noam. 2009. Turing on the 'Imitation Game'. In Robert Epstein, Gary Roberts & Grace Beber (eds.), *Parsing the Turing Test*, 103–106. Amsterdam: Springer.
Chomsky, Noam & George A. Miller. 1963. Introduction to the formal analysis of natural languages. In R. Duncan Luce, Robert R. Bush & Eugene Galanter (eds.), *Handbook of Mathematical Psychology II*, 269–321. New York: Wiley and Sons.
Church, Alonzo. 1936. An unsolvable problem of elementary number theory. *American Journal of Mathematics* 58, 345–363.
Church, Alonzo. 1941. *The Calculi of Lambda Conversion*. Princeton, NJ: Princeton University Press.
Cooper, Barry. 2012. The incomputable reality. *Nautre* 482, 465.
Davis, Martin. 1958. *Computability and Unsolvability*. New York: McGraw-Hill.
Davis, Martin (ed.). 1965. *The Undecidable: Basic Papers on Undeciable Propositions, Unsolvable Problems and Computable Functions*. New York: Raven Press.
Davis, Martin. 2004. The myth of hypercomputation. In Christof Teuscher (ed.), *Alan Turing: Life and Legacy of a Great Thinker*, 195–212. Berlin: Springer.
Davis, Martin. 2012. *The Universal Computer: The Road from Leibniz to Turing*. New York: Taylor & Francis Group.
Deutsch, David. 2011. *The Beginning of Infinity*. London: Allen Lane.
Evans, Nicholas & Stephen C. Levinson. 2009. The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and Brain Sciences* 32, 429–492.
Everett, Daniel L. 2005. Cultural constraints on grammar and cognition in Pirahá: Another look at the design features of human language. *Current Anthropology* 46, 621–646.
Fodor, Jerry A. 2001. Doing without what's within: Fiona Cowie's critique of nativism. *Mind* 110, 99–148.
Gallistel, C. Randy. 2010. Learning organs. In Jean Bricmont & Julie Franck (eds.), *Chomsky Notebook*, 193–202. New York: Columbia University Press.
Gallistel, C. Randy & Adam Philp King. 2009. *Memory and the Computational Brain: Why Cognitive Science Will Revolutionize Neuroscience*. New York: Wiley-Blackwell.
Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum & Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

Gleick, James. 2011. *The Information*. New York: Pantheon.
Gödel, Kurt. 1934 [1965]. On undecidable propositions of formal mathematical systems. In Martin Davis (ed.), *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*, 39–74. New York: Raven Press.
Hammeroff, Stuart R. & Roger Penrose. 1996. Orchestrated reduction of quantum coherence in brain microtubules: A model for consciousness. In Stuart R. Hameroff, Alfred W. Kaszniak & Alwyn C. Scott (eds.), *Toward a Science of Consciousness: The First Tucson Discussions and Debates*, 507–540. Cambridge, MA: MIT Press.
Hauser, Marc D. 2009. The possibility of impossible cultures. *Nature* 460, 190–196.
Hauser, Marc D., Noam Chomsky & W. Tecumseh Fitch. 2002. The faculty of language: What is it, who has it, and how did it evolve? *Science* 298, 1569–1579.
Hofstadter, Douglas R. 1979. *Gödel, Escher, Bach: An Eternal Golden Braid*. New York: Basic Books.
von Humboldt, Wilhelm. 1836 [1989]. *On Language: The Diversity of Human Language-Structure and its Influence on the Mental Development of Mankind*. Cambridge: Cambridge University Press.
Hume, David. 1763. *The History of England: From the Invasion of Julius Caesar to the Revolution in 1688*. Google eBook.
Jackendoff, Ray. 2007. *Language, Consciousness, Culture: Essays on Mental Structure*. Cambridge, MA: MIT Press.
James, William. 1890. *The Principles of Psychology*. New York: Holt.
Kearns, Michael J. & Leslie G. Valiant. 1994. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of American Academy of Mathematics* 41, 67–95.
Kleene, Stephen C. 1936. General recursive functions of natural numbers. *Mathematische Annalen* 112, 727–742.
Kleene, Stephen C. 1980. The theory of recursive functions, approaching its centennial. *Bulletin of the American Mathematical Society* 5, 43–61.
Kripke, Saul. 1982. *Wittgenstein on Rules and Private Language*. Cambridge, MA: Harvard University Press.
Lashley, Karl S. 1951. The problem of serial order in behavior. In Lloyd A. Jeffress (ed.), *Cerebral Mechanisms of Behavior*, 112–146. London: Chapman & Hall, Limited.
Leibniz, Gottfried Wilhelm. 1714 [1992]. Monadology. In Nicholas Rescher (ed.), *G.W. Leibniz's Monadology*, 45–308. Pittsburgh, PA: University of Pittsburgh Press.
Marr, David. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York: Freeman.
Mandelbrot, Benoit B. 1982. *The Fractal Geometry of Nature*. New York: W.H. Freeman.
McCarthy, John. 1956. Inversion of functions defined by Turing machines. In Claude E. Shannon & John McCarthy (eds.), *Automata Studies*, 177–182. Princeton, NJ: Princeton University Press.
Miller, George A., Eugene Galanter & Karl H. Pribram. 1960. *Plans and the Struc-*

ture of Behavior. New York: Holt, Rhinehart, & Winston.
Minsky, Marvin. 1963. Steps toward artificial intelligence. In Edward A. Feigenbaum & Julian Feldman (eds.), *Computers and Thought*, 406–450. New York: McGraw-Hill.
Minsky, Marvin. 1985. Why intelligent aliens will be intelligible. In Edward Regis (ed.), *Extraterrestrials*, 117–128. Cambridge, MA: MIT Press.
von Neumann, John. 1928. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen* 100, 295–320.
Penrose, Roger. 1989. *The Emperor's New Mind: Concerning Computers, Minds and the Laws of Physics*. Oxford: Oxford University Press.
Pinker, Steven. 1979. Formal models of language learning. *Cognition* 7, 217–283.
Pinker, Steven. 1984. *Language Learnability and Language Development*. Cambridge, MA: Harvard University Press.
Pinker, Steven. 1994. *The Language Instinct: The New Science of Language and Mind*. New York: Penguin.
Pinker, Steven. 2002. *The Blank Slate: The Modern Denial of Human Nature*. New York: Penguin.
Poeppel, David & David Embick. 2005. Defining the relation between linguistics and neuroscience. In Anne Cutler (ed.), *Twenty-First Century Psycholinguistics: Four Cornerstones*, 103–118. Mahwah, NJ: Lawrence Erlbaum.
Poeppel, David & Akira Omaki. 2008. Language acquisition and ERP approaches: Prospects and challenges. In Angela D. Friederici & Guillaume Thierry (eds.), *Early Language Development*, 231–253. Amsterdam: John Benjamins.
Polanyi, Livia & Remko Scha. 1983. On the recursive structure of discourse. In K. Ehlich & Henk van Riemsdijk (eds.), *Connectedness in Sentence, Discourse and Text*, 141–178. Tilburg: Tilburg University.
Post, Emil. 1936. Finite combinatory processes: Formulation 1. *Journal of Symbolic Logic* 1, 103–105.
Post, Emil. 1944. Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society* 50, 284–316.
Post, Emil. 1947. Recursive unsolvability of a problem of Thue. *The Journal of Symbolic Logic* 12, 1–11.
Postal, Paul M. 2004. *Skeptical Linguistic Essays*. Oxford: Oxford University Press.
Postal, Paul M. 2009. The incoherence of Chomsky's 'biolinguistic' ontology. *Biolinguistics* 3, 104–123.
Putnam, Hilary. 1961. Some issues in the theory of grammar. *Structure of Language and Its Mathematical Aspects: Proceedings of the Symposium in Applied Mathe-matics: American Mathematical Society* 12, 25–42.
Putnam, Hilary. 1975. *Mind, Language and Reality: Philosophical Papers,* vol. 2. Cambridge: Cambridge University Press.
Roberts, Ian. 2011. Parametric hierarchies. Ms., University of Cambridge.
Russell, Bertrand. 1948. *Human Knowledge: Its Scope and Limits*. London: George Allen & Unwin.
Simon, Herbert A. 1962. The architecture of complexity. *Proceedings of the American Philosophical Society* 106, 467–482.
Shor, Peter W. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* 26,

1484–1509.
Soare, Robert I. 2009. Turing oracle machines, online computing, and three displacements in computability theory. *Annals of Pure and Applied Logic* 160, 368–399.
Turing, Alan M. 1936. On computable numbers, with an application to the *Entscheidungsproblem*. *Proceedings of the London Mathematical Society* 42, 230–265.
Turing, Alan M. 1939. Systems of logic based on ordinals. *Proceedings of the London Mathematical Socidety* 45, 161–228.
Turing, Alan M. 1947 [2004]. Lecture on the automatic computing engine. In B. Jack Copeland (ed.), *The Essential Turing*, 378–394. Oxford: Oxford University Press.
Turing, Alan M. 1950. Computing machinery and intelligence. *Mind* 59, 433–460.
Turing, Alan M. 1952. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London, Series B, Biological Sciences* 237, 37–72.
Turing, Alan M. 1954. Solvable and unsolvable problems. *Science News* 31, 7–23.
Valiant, Leslie G. 1984. A theory of the learnable. *Communications of the American Academy of Mathematics* 27, 1134–1142.
Vaux, Bert & Jeffrey Watumull. 2012. The phonological Turing machine. Ms., University of Cambridge & MIT, Cambridge, MA.
Watumull, Jeffrey. 2010. Merge as a minimax solution to the optimization problem of generativity. MPhil Thesis, University of Cambridge.
Watumull, Jeffrey. 2012. The computability and complexity of generativity. Ms., MIT, Cambridge, MA.
Watumull, Jeffrey & Ron Eglash. 2011. Fractal syntax. Ms., MIT, Cambridge, MA & Rensselaer Polytechnic Institute, Troy, NY.
Watumull, Jeffrey, Marc D. Hauser & Robert C. Berwick. In preparation. Problems and solutions for comparative research on the evolution of syntactic computations.
Weinberg, Steven. 1976. The forces of nature. *Bulletin of the American Academy of Arts and Sciences* 29, 13–29.
Wolf, Ronald de. 1997. Philosophical applications of computational learning theory: Chomskyan innateness and Occam's razor. Master's thesis, Erasmus University Rotterdam.
Wolfram, Stephen. 2002. *A New Kind of Science*. Champagne, IL: Wolfram Media.
Zylberberg, Ariel, Stanislas Dehaene, Pieter R. Roelfsema & Mariano Sigman. 2011. The human Turing machine: A neural framework for mental programs. *Trends in Cognitive Science* 15, 293–300.

Jeffrey Watumull
University of Cambridge
Department of Theoretical and Applied Linguistics
Sidgwick Avenue
Cambridge CB3 9DA
UK
jw647@cam.ac.uk

Massachusetts Institute of Technology
Department of Linguistics and Philosophy
77 Massachusetts Avenue, Building 32
Cambridge, MA 02139
USA
watumull@mit.edu