

Indistinguishable from Magic: Computation is Cognitive Technology

John Kadvany

Received: 6 April 2009 / Accepted: 25 January 2010 / Published online: 10 February 2010
© Springer Science+Business Media B.V. 2010

Abstract This paper explains how mathematical computation can be constructed from weaker recursive patterns typical of natural languages. A thought experiment is used to describe the formalization of computational rules, or arithmetical axioms, using only orally-based natural language capabilities, and motivated by two accomplishments of ancient Indian mathematics and linguistics. One accomplishment is the expression of positional value using versified Sanskrit number words in addition to orthodox inscribed numerals. The second is Pāṇini’s invention, around the fifth century BCE, of a formal grammar for spoken Sanskrit, expressed in oral verse extending ordinary Sanskrit, and using recursive methods rediscovered in the twentieth century. The Sanskrit positional number compounds and Pāṇini’s formal system are construed as linguistic grammaticalizations relying on tacit cognitive models of symbolic form. The thought experiment shows that universal computation can be constructed from natural language structure and skills, and shows why intentional capabilities needed for language use play a role in computation across all media. The evolution of writing and positional number systems in Mesopotamia is used to transfer the thought experiment of “oral arithmetic” to inscribed computation. The thought experiment and historical evidence combine to show how and why mathematical computation is a cognitive technology extending generic symbolic skills associated with language structure, usage, and change.

Keywords Computation · Formal grammar · Positional value · Sanskrit · Pāṇini · Rewrite systems · Intentionality · Grammaticalization · Writing

J. Kadvany (✉)
Policy & Decision Science, 1070 College Avenue, Menlo Park, CA 94025, USA
e-mail: john@johnkadvany.com
URL: johnkadvany.com

The Problem

Modern computation exists—What makes it possible? Computation here includes all styles of computer and device programming, school arithmetic calculation, algorithm design, and alternative theoretical expressions of computation itself. Perhaps most humans have computed or enumerated in some way, using media including notched sticks, body-parts, speech, colored rods, knotted ropes, the abacus, electronic devices, and clay, sand or paper inscription. But the computational representations found in most civilizations fall short of “universal,” meaning the theoretical power of typical programming languages, many axiomatic systems, universal Turing machines, or many equivalents. Ancient Greek mathematics, while containing proofs by contradiction, and intricate geometric constructions, did not master general exponentiation, $c = a^b$, and Greek multiplication is wedded to measurement units.¹ In contrast, children trained to use positional numbers (i.e. in 3050, 3 is three thousand, 5 is fifty), have the computing skills of priests long ago.²

From the perspective of human cognition, arithmetic going beyond addition and multiplication—“+ and ×”—all the way to modern computation, represents quite an achievement. It’s not obvious how such symbolic processing is possible. Computations are expressed in language, however formalized, but even elementary methods for efficient multiplication or division are not implicit in the recursive patterns found in typical natural languages.³ Useful number words or symbols facilitate arithmetic of some scope in many languages and cultures. But, without some language change or expansion, we don’t find natural languages whose grammar (including phonology, morphology, syntax, semantics) can be implicitly used to represent much multiplication, not to mention behavioral and interpretative skills for utilizing new number words or symbols. So what cognitive skills make it possible to transcend the weak computing power of natural languages, and what relationships hold between linguistic and mathematical recursion, or the kinds of symbolic generativity possible in natural and mathematical language? Taking an ethological stance, we can ask what kinds of symbolic or cognitive skills make modern computation possible when our innate number skills are apparently limited to the perception of small “numerosities,” meaning collections of like objects or

¹ Fowler (1999, chap. 1) discusses Greek mathematics as “non-arithmetical” and (Unguru 1991; Fowler 1994) debate the scope of Greek induction, a close relative of recursion.

² For example, Egyptian mathematics used an additive number system which, through a clever trick, could also be used for reasonably efficient multiplications. But that construction, while effective and correct, is not efficient enough for much further algorithmic design. Similarly, one can use high school Roman numerals for multiplication, but they quickly become difficult to manipulate further. The great historical and cognitive solution here is positional value, whose compact linguistic expression in ancient India plays a key role below. See also note 25 below.

³ “It is in this sense than an arbitrary Turing machine, or an unrestricted rewriting system, is too unstructured to serve as a grammar. By imposing further conditions on the grammatical rules, we arrive at systems that have more linguistic interest but less generative power”(Chomsky 1963, 359). Hence natural language recursion, however complex, is still computationally weak; see also (Pullum and Scholz 2005; Chomsky 1980, 123).

perceived patterns (Dehaene 1997). Lacking needed leverage, we could just be stuck with addition and some bounded multiplicative algorithms.⁴

Franz Boas once remarked that, though the complex languages of North America Indians often lacked number words beyond small values, he could easily introduce additional number terms, and their use, to speakers as new grammar.⁵ “Primitive” language therefore was a red herring, and Boas’ student Edward Sapir argued that all languages were, roughly, functionally equivalent.

Implicit in these observations is a capacity for *extending* a language to a new one. Whether that is advantageous or desirable is a contingent cultural value; but apparently less so are possibilities for linguistic and cognitive transformation. Uncoincidentally, Sapir was a great analyst of language change, saying even that languages, including their syntax and semantics, were always in flux, just usually unconsciously and not on temporal or geographical scales perceived by individual speakers (Sapir 1921, 144). The first goal of this paper is to show that ordinary language change, including the cognitive skills making it possible, is typically sufficient for increasing the computational power of natural language to that needed for universal computation. In a nutshell, mathematical recursion can be constructed from linguistic recursion through the same intentional skills found in language use, but oriented to the creation of artificial languages needed for arbitrary computations and hence modern mathematics.⁶ The paper’s second goal is to map processes occurring in speech, and in natural language change, to parallel processes for computation in written or other media. For all this, we need means for thinking about formal computing languages in terms of natural language constructions and their transformation.

A Thought Experiment

Suppose an energetic Boas introduces into, say, spoken Nootka or Haida, in addition to some number terms and arithmetic, an entire computational scheme, sufficient to represent any axiomatic system, algorithm, or computing idiom you please. Such a generative system would be as good as any for producing potentially infinite sets:

⁴ I focus on the transition from additive to multiplicative algorithms because: (1) additive number systems, while not universal across languages, are common and easy to construct in many grammars, with number words coordinating, e.g., body-part counting and other one–one enumeration behaviors, possibly sufficient for *bounded* multiplication; (2) in formal models of arithmetic, addition and multiplication (but not addition alone) are sufficient to represent universal computation; (3) with multiplication and universal computation, *intensional* phenomena occur in the formation of consistency statements (Boolos and Jeffrey 1979, 186), and these are a symptom of, but not identical with, *intentionality* (Searle 1983, 24).

⁵ Boas and Powell (1911). More recently, (Everett 2005) controversially argues that the Amazonian Pirahã language lacks recursive syntax in ways challenging Chomsky’s claim that recursion is a linguistic universal; Everett also conjectures that a stringently local and finitist worldview makes some abstractions and syntactic forms mostly unnecessary. My approach is a complement: *given* certain types of linguistic recursion, whether universal or not, *then* one can construct mathematical recursion from it as discussed below. As with Everett, the material and cognitive setting is all important to the development of grammatical form.

⁶ See (Dehaene 1997, chap. 9) on cognition and modern mathematics; the present paper addresses some of the issues raised there.

distant digits of pi, the computing programs in some idiom, algorithm calculations for a potentially infinite set of inputs, consequences of axiomatized theories, logical truths, grammatical fragments of natural languages, planetary positions and tidal flows, numbers of mating rabbits, a traveling salesman's shortest tours.

We allow that the fictive Boas might know his logic from written sources. But Boas and the speakers whom he instructs, and whose language is being expanded, use no writing or kindred inscription in a significant way. They construct an *oral arithmetic* in natural language: a heard and recited computational system able to represent, and compute with, any algorithm you like, just like a modern programming language. Computations, in addition to needed definitions, axioms and rules, are also carried out orally, limitations on time and memory notwithstanding—a standard assumption in computational theory. For this thought experiment to work means being able to introduce notions of a finite discrete symbol set on which the whole construction is based, conveniently organized in finite lists or matrices; recursive rules applied to symbol sets to define categories like numerals, data types, constants, variables, formulas, terms, equations, programs, proofs, and computations; and sufficient flexibility to recursively combine and modify algorithms in typical ways for achieving computational goals. Thus would Boas' students' create a computational space equal to that of a universal Turing machine or its many equivalents, including most modern computing languages.

What might such a thought experiment show, assuming sufficiently realistic steps to carry it out? First, that advanced mathematical computation is a cognitive technology which can be constructed using the same resources by which new language patterns get constructed from old. Language change occurs through communicative usage, so the requisite skills include capabilities for coordinating grammatical forms with attention, mental models, common knowledge, and communicative intentions generally. Here these skills would be directed to counting and algorithmic behavior, especially the manipulation of computing symbols themselves. We assume these intentional skills, as far as they play a role in language change, to be just as necessary for our imaginary Boas as for language use generally.⁷ And language change is no fiction, but a well-documented phenomenon occurring across many language families (McWhorter 2001). Modern English grammar uses word order to distinguish syntactic roles in *man bites dog* from *dog bites man*, but in Old English the latter can be either *canis hominem mordet* or *hominem canis mordet*, with grammatical roles carried by word affixes. Over centuries of changing usage and contact with other languages, the intricate Old English case system eroded as sound changes rendered many inflections and word endings inoperable. Functional roles of the old syntax were gradually replaced, mostly unconsciously over generations, in a variety of ways. New grammatical function words emerged, like *will* as an “auxiliary” to mark future tense *I will ___*, when it earlier had only ordinary meanings of capability or skill; similarly for the auxiliaries *can*, *should* and *must* (Barber 2000). Then many case endings were supplanted by generically useful prepositions like *of*, *in*, *with*, *by*, *for*, which keep

⁷ See (Clark 1996; Tomasello 1999, 2003) on language use and the coordination of intentionality.

words separate in ordered relations instead of conjoining words to marking affixes, the latter enabling free word order. The development of new function words, along with new roles for word order, are all good examples of *grammaticalization*. The syntactic differences between Old and Modern English could be greater—tones and clicks, for example, were never adopted to mark syntactic roles—but the structural changes are dramatic and representative of many other examples. Empirical language change extends far into language structure, well beyond “mere” sound change, vocabulary modifications, or dialect formation. So in our thought experiment for oral arithmetic, we assume typical cognitive skills making comparable linguistic innovation possible. Assuming sufficient realism for our thought experiment, computation then is similarly possible as an artifice of language and mind.

For oral arithmetic, we further assume that grammatical change is motivated by some domain-relevant cognitive model. All kinds of perspectives get marked through a language’s patterns of affixing, inflection, tones, or word ordering. If the need, conscious or not, is to mark control, or broadly construed “ownership,” that may be indicated by a grammatical possessive—*the cat’s anger, my left foot*; if the cognitive function is a conception of time or its passage, then the marking is of tense or aspect; if there is motivation to indicate the quality of knowledge the speaker has of a topic, then, as in modern Turkish, “evidentiary” marking can indicate direct, indirect, or hearsay information; if a specialized conception of direction and place is useful, then spatial syntax can be used, as in some Mesoamerican languages; or distinctive physical relationships can be indicated, as in Korean prepositions indicating “tightness of fit.” Thus all kinds of folk knowledge or heuristic models of the life world get grammaticalized. The total grammaticalization of arithmetic, mathematics, and logic in our Boasian thought experiment may be unusual, but all that really differs is the role of counting and symbolic manipulation as the relevant functional domain.⁸

The many exotic examples of grammaticalization from the world’s languages do not entail radical differences in sensory perception, nor that various cognitive perspectives are either determined as either necessary or impossible.⁹ Depending on the language involved, relevant perceptual tasks using different languages may be easier for fluent speakers, and thought correlated with notions codified in the grammar can be, all else being equal for the language user, easier to formulate, express, or communicate. The various options for a language’s syntax means users need to notice the presence or absence of marking correlates in the world. Concepts expressed directly in grammars—evidence for assertions, spatial marking, time and aspect, physical patterning paradigms, etc.—conditions users to process

⁸ Number systems abound in history which extend counting behaviors, especially one–one enumerations, through number words and their productive rules. But not found are non-written linguistic examples with much computing power beyond simple addition or very limited multiplication, and so these systems don’t reveal much about cognition for modern computation and mathematics.

⁹ Gentner and Goldin-Meadow (2003) revisit the Sapir-Whorf hypothesis, not generally thought of as including writing and computation (Whorf 1956). While the thought experiment in the text shows that, in principle, linguistic change can lead to language in which all computations are possible, in reality the change relies on our perceptions of inscribed computations. So for computation, the increase in conceptual power depends critically on changing media to realize the more intricate symbolic apparatus.

information differently, and measurable outcomes, while neither radically creative nor limiting, are demonstrable.¹⁰

But there is a difference with number language. Even simple arithmetical tasks are sufficiently complex that much progress is virtually impossible without specialized syntax and semantics. Mathematical knowledge is a case where the cognitive limits are a real obstacle, not just an inconvenience or inefficiency. Some barriers to the recursive constructions needed for computation are overcome by inscription or other prosthetics (e.g. an abacus), which change the modality for symbolic perception and relax memory constraints on symbolic processing. That shift in media is essential, but it also occludes the construction of stronger computational language and qualitative differences in what can be counted, structured, and perceived in numeric or logical terms.¹¹ Hence the need for the oral arithmetic thought experiment, in which the benefits of inscription are temporarily set aside.

This need to bypass inscription, and the role for changing media, is already suggested by modern logic. Formal systems are usually thought of as being written, but no mathematics should depend on that, at least directly, just as geometry should not depend on its realization as perceived objects in perceived space. Inscription, mathematically, has to be a dispensable heuristic metaphor.¹² Mathematical computation can occur *in* what media you like *of* whatever discrete symbols you like. Formally, the construction of oral arithmetic involves two languages, L and L' , the latter a transformed version of the former with greater computational power: in L' elements of a formal system can be listed, new terms recursively defined, and rules applied to program, prove, or otherwise compute in ways not possible in L . The challenge, for the thought experiment, is that while language change can involve radical differences in structure, expressive power, or efficiency, that doesn't imply significant jumps in mathematical strength. The puzzle is to create that increment of computing power, respecting an "experimental" constraint for the transformation $L \rightarrow L'$ limiting us to known features of language change, principally grammaticalization and allied cognitive skills. Following that, we can more realistically consider similar changes realized in writing rather than speech. In this way, natural language change becomes a model of symbolic computation.

The next step is to see in detail how the recursive constructions needed for universal computation can be expressed in natural language.

¹⁰ See (Slobin 2003) on "thinking for speaking"; (Dehaene 1997, 102) on counting skills for native Chinese speakers; and (Levinson 2003) on spatial marking.

¹¹ See (Bloom 1994) on the transfer of recursion across cognitive domains; my approach focuses on external media as perceived and leveraged by fixed cognitive capacities.

¹² For example, in logic, "alphabet" and "rewrite" should be metaphors whose mathematical content has nothing to do with inscription per se, just as Plato advised that geometry "...is in direct contradiction with the language employed in it by its adepts...they speak as if they were doing something [practical]... their talk is of squaring and applying and adding and the like, whereas in fact the real object of the study is pure knowledge" (*Republic* 518d, 527). Difficult though, is understanding what an abstract discrete symbol is outside of any language or semiotic system at all; see "Grammaticalization in Script" below.

Constructing an Oral Formalism in Ancient India

A “programming,” “formal,” or “symbolic” system is defined by a finite set of discrete symbols combined according to precise recursive rules for generating a potentially infinite target set of symbolic expressions. We leave open the computational power of the rules, and the mathematical content of generated expressions, but have in mind systems like modern programming languages or computing paradigms which can be tailored to various uses. Remarkably, nearly all the intellectual technology needed for a formal computing system built directly from natural language resources has existed for over two thousand years. The first formal system, before Gottlob Frege’s 1879 *Begriffsschrift* for first-order logic, is Pāṇini’s approximately fifth-century BCE Sanskrit grammar. Unlike modern systems, Pāṇini’s grammar is defined for *oral* recitation. Its finite symbol set consists of Sanskrit phonemes, word stems, and roots; these form the system’s inductive basis, to which are applied explicit rules for generating a potentially infinite set of target expressions of classical Sanskrit. While having no mathematical content, the system uses many techniques essential to modern logic and computation.¹³

Because knowledge of this achievement is familiar mostly to Sanskritists and linguists, attention has to made at the outset to this historical lacuna. Pāṇini is indeed *the Indian Euclid*: his interest is in language rather than number or shape; his expertise is formal recursive generation rather than axiomatic deduction; and his habitus is the oral world of Sanskrit speech, the language of exact science in ancient India (Staal 1988, 2006). Pāṇini’s grammar is mentioned as a generative ancestor on the first page of Chomsky’s *Aspects of the theory of Syntax*, and was lauded by Leonard Bloomfield as one of the greatest intellectual accomplishments ever (Bloomfield 1933, 11), its rules being codified as about 4,000 mnemonically efficient aphorisms formulated as versified Sanskrit sūtras; these define the “programming manual,” so to speak.¹⁴ From a linguistic perspective, the rules are purely descriptive with no intended psychological meaning, and are organized generally through “tiered” constructions coordinating semantics, syntax, morphology, and phonology, much as in modern linguistics.

For our thought experiment, what’s relevant is Pāṇini’s complete mastery of formal language definition, using an extension of Sanskrit as a metalanguage for

¹³ While modern generative linguistics borrowed formalist ideas from mathematical logic, ancient Indian linguists mostly developed them directly, building on basic ideas of recursive system construction developed to exactly describe generatively patterned rituals, with portions of the early Vedas known as “ritual manuals” (Renou 1941, Staal 1990). Included here are the earliest phonological theories and the segmentation of continuous recitation into discrete units (*saṃhitā* vs. *padapāṭha*) for analytical purposes (Staal 2006, 77). On expressions of generality and the potential infinity of language generated by finite means in Indian grammatical theory, see (Staal 1990, 89).

¹⁴ “These sūtras are like nothing so much as the rules in a computational grammar of a modern language, such as might be used in a machine translation system: without any musical or ritual element, they apply according to abstract formal principles. This is not a metaphor, or anachronistic interpretation of Sanskrit grammar, but a straightforward description of the working of the sūtras in Pāṇini’s system” (Ostler 2005, 181). This is a slight overstatement because of initializing *kāraka* rules, requiring user-based assignments of categories of “agent,” “patient,” etc.; see (Gillon 2007).

spoken Sanskrit as the object language. The extension involves only constructions already appearing in Sanskrit. Pāṇini's rules use special sounds, affixed to expressions being constructed, as markers indicating grammatical roles or linguistic constraints. Among other techniques, these phonemic signs control repeated "rewriting"—really "respeaking"—so that word stems and roots, with which linguistic derivations begin, are transformed into genuine Sanskrit expressions.

For example, to derive *devadatta is cooking rice in a pot for yajñadatta* (*devadatta odanam yajñadattāya sthālyām pacati*), *devadatta* is to be marked as a singular agent who is neither the speaker or hearer of the utterance; *cook/pac* is the verbal root of action and must be in agreement with the agent; *rice/odana* is the patient of the cooking action; *yajñadatta* is the beneficiary of cooking; *pot/sthāli* is the cooking location. That semantical information, set by the user, is marked by affixes to give *devadatta + sU*, *pac + LAT*, *odana + am*, *yajñadatta + Ņe*, *sthali-Ņi* (Gillon 2007). The capitals represent "indicatory" or auxiliary sounds, called *IT* (from the Sanskrit particle *iti* used for oral quotation), and identify relevant syntactic information; in modern terms, these "non-terminal" auxiliary symbols control the correct production of "terminal" Sanskrit speech sounds. The auxiliaries ultimately are deleted in the derivation with other sounds (here in lower case) combined with the initial verbal stems and nominal roots. *IT* markers are used in clever ways to define oral lists and matrices, and are also used to define categories, such as derived compounds, which then are recursively referenced for further compounding or other uses. Many phonological *sandhi* rules, for fusing sounds across word or affix boundaries (the *s* in *devadatta + sU*), are rigorously context-sensitive, reflecting local constraints for rule application. Pāṇini's use of auxiliary markers is completely systematic, repeated in dozens of rules and metarules (*paribhāṣā*) defining system use. The grammar characterizes linguistic case agreement, anaphor, the formation of complex compounds, word order, and much else.¹⁵

Pāṇini's oral system follows a contemporary standard of computing design: the method of *rewrite rules* studied by Emil Post in the 1930s. A Post production system is defined by a finite alphabet of symbols, a finite set of axioms over the symbol set, and a finite set of rules which produce target expressions through repeated application starting from the axioms (Davis 1965, 288; Minsky 1967, chaps. 12–13). For example, starting with symbols $\{a, b\}$, the rules $\{X \rightarrow aXa, X \rightarrow bXb\}$ generate two-letter palindromes starting with axioms $\{a, b\}$ and X an auxiliary "variable" symbol standing for any expression, possibly empty. Post showed such systems in full generality could be used to represent any Turing machine construction, and therefore had universal computing power: any effective procedure or algorithm, or the expressions generated by any typical axiomatic system, can be replicated by a suitable alphabet and Post production rules.

Post's *proof* of the scope of his systems is directly relevant to Pāṇini's method of auxiliary symbols. Post showed that the formal language L_P generated by any production system P could be replicated using rules and axioms in a standard format ($gX \rightarrow Xh$, with g and h fixed strings), using *additional* auxiliary symbols to control

¹⁵ For introductions to Pāṇini's grammar see (Gillon 2007; Sharma 1987) On the evolution of metalinguistic concepts in India see (Staal 1975).

the exact production of L_P . Post showed in a canonical way how to construct systems P^* and metalanguages L_{P^*} suitable to generate exactly any object-language target L_P . Pāṇini's target was ancient Sanskrit, while Post's was recursively enumerable sets of natural numbers; both faced the challenge of devising a finite set of replacement rules to exactly reproduce various series of potentially infinite patterns. Post's key idea, similar to what was exploited by Pāṇini and his predecessor linguists, is the expansion of the initial symbol set and addition of new axioms and metarules to control production of the object language. Post showed Pāṇini's procedure was a general one, and that rewrite systems have the same algorithmic power as Turing machines or any other standard computational model. So it makes perfect mathematical sense that Pāṇini repeatedly exploited the method of auxiliary symbols, and he probably saw that he could construct any rule he needed to describe useful grammatical categories, expression forms, or arcane linguistic constraints. Historically, it's appropriate to consider Post as rediscovering and generalizing Pāṇini's technique in the context of modern mathematical logic and early modern theories of computation.

It follows that if a grammarian, like our fictive Boas, can define rules using the method of auxiliary symbols, and apply them in "affixing" patterns $gX \rightarrow Xh$, he can in principle reproduce the expressions defined by any Post production system. The same holds therefore for Pāṇini's grammar taken as a paradigm for system construction, and expressed orally using Sanskrit sounds, stems and roots as its "formal symbols." Pāṇini not only created a formalism, he tacitly discovered how to represent any typical formal system with a finite basis and recursively defined rules. The remarkable accomplishment of Indian linguistics was to see that the devices needed to accomplish that could be created via Sanskrit grammaticalization.¹⁶ Pāṇini's grammar shows that formal systems of virtually arbitrary computational or axiomatic power can be directly constructed from mastered speech. Such artificial languages, in principle, are a continuation of natural language grammar by its own means, and their computational power should be understood as extensions of otherwise typical natural language skills. Hence the oral arithmetic thought experiment is: add to Pāṇini's grammar additional rules, in sūtra form if you like, codifying whatever computations you like, using the Pāṇini-Post method of auxiliary symbols and rewrite rules, then applying rules in recitation, perhaps as instructions for dustboard calculations. For the thought experiment, all the basic computing technology is in place. All that is missing are axioms and rules to generate target computations of contemporary interest, perhaps codified using additional sounds to demarcate "proofs," "computations," or "number statements" in the extension $L \rightarrow L'$. All that differs from more typical language change is that

¹⁶ It was essential that Pāṇini's artificial language extend Sanskrit in a natural way for its use would otherwise be seen as polluting, or at least vulgar, given the privileged status given language (*vac* or *brāhman*) in Vedic culture. Pāṇini's grammar was meant to ensure exact oral reproduction across generations because the linguistic expression itself, when a correct copy of the original, was sacred, the voices of men speaking the language of gods. But pragmatically, Sanskrit was also intended as an efficient and accurate means of oral communication across a huge land, and the grammar provided needed consistency. Hence the grammar is descriptive but was taken as prescriptive for a combination of metaphysical, pragmatic-communicative, and socio-political reasons (Staal 1995).

the modifications here are deliberate, reflective, and conscious, none of which eliminates the intentional skills needed for constructing this computational system; if anything, just the opposite.

Now, in spite of such a quick route to oral arithmetic, the approach is unfortunately distanced from numeric representations and the transition to higher computation. Pāṇini's target is language, not mathematics, even if his rules can be contrived to apply to numbers. More useful would be specifics of how a linguistic number system gets an increase in computational power. As it turns out, such a bridge also exists between Sanskrit and modern mathematics, via the Indian method of positional value, their world-historical contribution to all subsequent quantitative reasoning. True to the standard of oral Sanskrit as the *sine qua non* for scientific expression, the Indians developed a spoken system of *positional number words*. The conjectured development of these Sanskrit expressions provides an example of linguistic grammaticalization involving one of history's most powerful and ingenious computational devices.

Perhaps as early as 200 BCE, Indians knew the positional principle, either learned elsewhere or discovered themselves (Datta and Singh 1935). Unusually, the earliest extant expressions are positional number words, not traditional number signs, making possible representations of very large numbers in sūtras involving cosmological theories, astronomical events, or special calendrical values, such as ritual start days. For example, the number word *śāra.yama.rasa* (“.” to separate words) is composed of *śāra* (= 5, as in five arrows of Kama, god of love), *yama* (= 2, “a pair”) and *rasa* (= 6, for the six tastes); the value 625 (read right to left) is defined by the standard positional rule. The symbol set for these decimal representations consists of large finite set of names for each of $0 \rightarrow 9$ (or $1 \rightarrow 9$ before the zero); e. g. two can be *yama*, *aśvin* (twin sons of the Sun), *netra* (the eyes), *kara* (hands), *pakṣa* (wings), etc. The many options make it easier to create euphonious and memorable expressions for any represented value, especially large ones: e.g. *agnisūnyāśvivasarpārṇava* names 488,203 and *vasvagniyamāśviśikhidasra* names 232,238 (Ifrah 2000, 411). So defined, positional number words are not unique, but are completely unambiguous. The number words are Sanskrit compounds, perhaps the central generative form of the language (Williams 1846). Indeed the compactification achieved by the decimal positional rule—just n linear places encode 10^n values—epitomizes the extreme concision characteristic of Sanskrit verse. Pāṇini's grammar, devised before the appearance of positional number words, can easily be used to define these new compounds, including the use of multiple names as synonyms, so the multiplicative number words, intended for oral delivery, are definable in an orally defined formalism. Thus we have a real example of a significant increase in numeric representation, directly expressible in an oral formalism whose artificial language is just more natural language.¹⁷ Since our focus

¹⁷ Positional value can be defined by a schema for concatenated symbols $a_1 \dots a_{n+1}$ such as $pos_n: a_1 \dots a_n$
 $+1 = (10 \times a_1 \dots a_n) + a_{n+1}$. These equations can be defined in formal arithmetics for addition and multiplication, as can a single master formula $Pos(n, S)$ from which all pos_n can be derived for a finite symbol set S . In fact, addition alone (e.g. the complete and decidable formal theory Presburger arithmetic) is sufficient to individually define each pos_n , but a formula $Pos(n, S)$ involves general multiplication and so cannot be defined using addition alone. Simple formalisms including addition and general

is on just such transitions, we proceed with a linguistic conjecture about how this new symbolic machinery was devised, and what it achieves.

In 1863 Franz Woepcke, a prolific historian of Arabic and Indian mathematics, suggested that Sanskrit *names for powers* could be the means by which simpler *non-positional* number words were grammaticalized into positional number words. Names include *padma* (10^9), *kharva* (10^{10}), *nikharva* (10^{11}), *mahāpadma* (10^{12}), and assorted special names above 10^{17} , such as *asankhyeya* (10^{140}), meaning “innumerable.” As with names for $0 \rightarrow 9$, individual powers can have numerous names. Woepcke noticed that *all* powers to 10^{17} were named in Sanskrit. Therefore for values below 10^{18} , number words could be formed in which there were no “skipped” powers, as we have in English with *millions*, *billions*, *trillions* (10^6 , 10^9 , 10^{12}) increasing powers by three. Coefficients in English therefore require multiplicative hybrids, like *two-hundred and thirty million*. Instead, in Sanskrit, the consecutive named powers imply coefficients can be limited to names for $0 \rightarrow 9$, at least below 10^{18} . So a *non-positional* number word can be formed akin to **9** and **3** *daśa* and **4** *sāta* and **2** *sahasra* for 9342 (= 2,439), with *and* (Sanskrit *ca*) left out in the word compound. With many alternative names for powers possible, no name is essential, and its information is carried exactly by its perceived position in the number word. Hence the named powers can drop out, leaving the positional pattern to define higher units. With a *last* defined unit, say trillions, one is stuck at a bounded level of polynomial representation using a highest fixed power. The fate of most non-positional number systems, like the *-illions* model, is failure to recursively automate the formation of higher number units, exactly the benefit of positional grammaticalization.¹⁸ The grammar of Sanskrit *non-positional* number words, using consecutively named powers combined with many alternatives for those names, plausibly defines a perceived gestalt of the required symbolic form for positional value. The new construction automates the otherwise “manual” formation of consecutively higher powers.

Linguistically, position as a grammatical marker plays some role in perhaps all languages; even for largely free order Sanskrit, position determines meaning *inside* many compounds, so here too the linguistic resource is entirely orthodox. The motivation to compactify in Sanskrit is paramount, and creative compounding is ubiquitous; hence positional number words are, in Sanskrit, a natural grammatical formation. The new pattern requires only ordinary linguistic skills, but motivated by a mental model of symbolic perception, rather than more typical grammaticalizations

Footnote 17 continued

multiplication are sufficient for universal computation and metasymbolic operations generally, leading to incompleteness and undecidability results, including the intensional unprovability of consistency (cf. note 4 above). The two formal theory classes, of additive and multiplicative arithmetics, are useful correlates for the changes $L \rightarrow L'$ discussed in the text (Kadvany 2007).

¹⁸ The problem of automating number units was expressed by Archimedes in *The Sand Reckoner*, a tour de force in which he used a geometrical model to estimate the number of “atoms” in the universe as a *myriad-myriad units of the myriad-myriad-th order of the myriad-myriad-th period*, or $[(10^8)^{10^8}]^{10^8}$ (Dijksterhuis 1938); a *myriad* is 10^4 and a *myriad-myriad* is 10^8 . Cognitively, positional value, not discovered by Archimedes or other Greek mathematical giants, is a principal means by which algorithms using multiplication and exponentiation are easily formed, as described by Fowler in note 25 below. On “automation” as a product of grammaticalization see (Bybee 2003).

of time, possession, evidence, physical pattern, or other non-symbolic notions. Most unusual in the positional grammaticalization is the use of symbolic position as a numeric *parameter*—the first, second, etc. place—in the formation rule to create higher units, perhaps a unique example of that among the world’s languages. In the terminology of modern linguistics, the named powers are perceived as grammatical “slots” which could be filled by any name appropriate for that position, with the generalizing linguistic construction being added to the user’s mental lexicon.¹⁹ Sapir says that *all grammars leak*, which here means the opportunity to reinterpret non-positional patterns and eliminate their redundancy. Of course, it is likely that positional value was known through writing, and the number words are what remain of that low status medium in Vedic culture.²⁰ Woepcke’s proposed grammaticalization may be mostly hypothetical and unrepresentative of the ultimate positional source. But whether the positional concept originated directly in Sanskrit doesn’t matter greatly for us. Positional number words are fully consistent with Sanskrit grammar overall, especially given the hegemony of Pāṇini’s rules, the “predicate logic” of its day. Pāṇini models compounds in detail, and the hypercompact number words are just further excellent examples of a preferred linguistic paradigm motivated by legitimate communicative needs for efficiency and precision in the oral register.

The positional number words, along with Pāṇini’s computing language, provide means for carrying out the oral arithmetic thought experiment in *numeric* terms: define numbers via positional number words, including synonyms, and then add arithmetical axioms for $+$ and \times , or equivalent rules as desired. It’s as if one could speak using an abacus and a programming language. The Sanskrit number words show that the typically additive relations expressed in natural language, in which multiplications are possible but bounded by largest named units, can be extended to an unbounded multiplicative form using natural language grammar *and* intentional skills needed to formulate the positional rule through a cognitive model of symbolic form applied to an existing language and its use. The positional grammaticalization varies from Pāṇini’s auxiliary symbols, which are added to Sanskrit as new affixes obeying case-like rules. Instead, redundant slots become functional place-holders identified by counting across symbol locations. As mentioned, this positional count is a variable parameter of the grammatical rule which bootstraps simpler additive arithmetic into multiplicative position. While insufficient for universal computation—for that one needs $+$ and \times in full generality and hence some axioms and logical rules—the step is a neat transition from a weaker recursive linguistic pattern to a stronger mathematical one. The positional number words approximate, but do not quite define, full multiplication. Nonetheless, Pāṇini’s methods and the positional number words illustrate how an oral arithmetic can be grammaticalized from ordinary speech, using almost entirely historical data. All that is needed are axioms,

¹⁹ See (Jackendoff 2002) on the generative power of grammatical constructions making limited assumptions about syntactic structure; hence construction grammar does *not* imply “non-generative.”

²⁰ See (Goody 1987, chap. 4) on the likely role of writing in the design of Pāṇini’s complex matrix of phonemes, the *Śivasūtras*, and many other observations on the difficulties of expressing formal concepts (lists, tables, multiplication) in oral cultures. For other indicators of written influences on Pāṇini see (Datta and Singh 1935, I 18, 33).

easily represented in Pāṇini's grammar, to enumerate *all* proofs, *all* computations, *all* digits of pi, *all* sentences or phrases of a certain type. That final step is a truly modern one, understood only in the 20th century, in which a potentially infinite set is generated by execution of a single master rule, such as an enumerating universal Turing machine. Except for that, all details relevant the thought experiment of oral arithmetic are grounded in ancient Indian linguistics and mathematics.²¹

To summarize, we have:

- a. Pāṇini grammar as a fully formal, but oral grammar, grammaticalized from Sanskrit via new affixing rules;
- b. A simple hypothetical extension of Pāṇini grammar to “oral arithmetic” sufficient for universal computation;
- c. Sanskrit positional number word compounds grammaticalized from non-positional number words, the latter being found or easily formed in many languages.

Pieces (a) and (b) give us oral arithmetic directly, but lack a more organic connection to historical mathematics; yet it's still all just grammaticalization through Pāṇini's affixing technique. With (c), the construction is more direct between number and language, and could be directly coded in (a) or (b), even in ancient times; but positional notation alone, while multiplicative, is not equivalent to general multiplication and so falls short of universal computation. So, while not completely necessary, the “best” oral arithmetic can combine elements from all of (a–c). The ensemble provides a robust idea of oral arithmetic constructed from natural language resources via grammaticalization and the intentional skills making such processes possible, primarily reflection on language structure itself.

As marvelous as Indian linguistics and mathematics are, they are here only means to our next goal of identifying the intentional skills needed for modern computation.

Computation as Representational Redescription

In oral arithmetic, increased computational power comes about via grammaticalization, either via Pāṇini's affixes and categorical constructions, or the generalization of the positional number word rule, or some combination of both. However achieved, the increase in precision and computing power depends on the intentional recognition of symbolic form, including the decomposition and redescription of mastered language or numeric representations into new patterns. This capacity to manipulate symbolic patterns so well is an expression of fully

²¹ The bridges existing today between mathematics, generative linguistics, and computation did not quite exist for Indian linguistics and mathematics, even as each exploited aspects of algorithmic method unified in the twentieth century. In India, linguists used powerful algorithmic formalisms, but applied to language, not mathematical algorithms; the formalisms were not needed by mathematicians, who nonetheless made algorithmic design a key heuristic tool and method of expression, as opposed to proof structure. Shared features of linguistics and mathematics included the use of Sanskrit sūtras to describe algorithms or linguistic rules; and Sanskrit number words to codify positional notation using compounds ratified, in turn, by Pāṇini's linguistic rules.

“triadic,” or “mindreading,” cognitive skills, argued by some to be fundamental for language acquisition, some types of social interaction, and most importantly, the production and comprehension of communicative intentions via symbolic behavior.²² “Triadic” refers to the coordination and sharing of intentions—that’s at least “two” or a dyad—to use a “third” symbol or symbolic pattern, simply to refer, or to achieve general communicative goals. Mindreading, put simply, refers to facility in processing triadic representations, ranging from the simplest types of symbolic reference to complicated communicative intentions involving deception or irony.

In our context, the triadic nexus of signs and intentions involves alternative semantical valuations, like “the 1 in 105 is one hundred, but ten in 510,” similar to now well-worn false belief assessments like “Sally thinks the box has a candy in it, but she is wrong because I saw someone take it away.” Such processing involves not just mapping between symbols and world, but comparisons of multiple representations against some standard: Sally’s prior beliefs, the observer’s belief of what Sally missed and her current goal, and knowledge of the candy box itself. Similarly, composite number symbols require multiple identifications of constituent signs and their positions, and combining these in a calculation leading to the intended value. Many simple arithmetical tasks, including the use of positional value, involve metalinguistic descriptions—symbol, list, positions, start, end—which also rely on mindreading skills to coordinate symbol usage, especially regarding the production of a potential infinity of expressions. More generally, mindreading skills are essential to artificial language construction because they make possible *representational redescrptions* (Karmiloff-Smith 1992) of one language pattern into another, transformations $L \rightarrow L'$ as put above. These changing symbolic patterns are objective indicators of underlying intentional capabilities, whose reflective complexity enables increases in computing power, much as occurs with positional grammaticalization.

Representational redescription *first* includes mastery of some existing symbolic form, like non-positional notations, or some given level of language structure. In contrast to basic competence, in which representations are manipulated holistically—like use of a tourist dictionary—mastery means that perceived forms can be conceptualized in terms of component “parts”—starts, ends, repeated patterns, coefficients, positions—and these parts can then be interpreted in alternative roles, much like Sally’s beliefs. *Second* is the redescription of such symbolic parts and functional roles, when possible, into a new format for accomplishing some representational, communicative, or for us, computational, goal, achieved by repackaging the decomposed parts into a new symbolic form: a new word ending or inflection, a role for word order, an arithmetic manipulation, heuristics for interpreting jokes or irony.

In childhood language acquisition, representational redescription occurs through overlapping mastery of phonemic inventories, morphemes, phrases and discourse styles. Or, for Pāṇini, ordinary Sanskrit is first sufficiently mastered, and after that it

²² See note 7 above, as well as (Baron-Cohen 1997) on mindreading and triadic representation. With others, Baron-Cohen considers the inability to process triadic representations a critical factor in some forms of autism, an important observable correlate of the cognitive abstractions.

can be perceived as made up of words, word parts, syllables, and sound forms, along with all kinds of regular and irregular structure. Such learning processes depend on the child, or linguist, encountering a world in which languages already exist. For numeric invention or learning, the prerequisite forms include computationally weak arithmetic symbolism, itself a product of earlier historical development. In the case of language change, social groups respond to existing languages to which they are exposed, but responses are far slower than in language acquisition and with different types of behaviors. Thus representational redescription is a spectrum phenomenon, involving individual and social learning, and different kinds of symbolic change. These changes vary in their associated level of conscious awareness, with arithmetic and linguistic innovation requiring the most explicit expressions of the change to occur. As a final illustration germane to computation, *pretend play* of children “involves the violation of veridical description of reality as well as the manipulation of explicit representations of agents, primary features of play objects, and decoupled representations of those objects in their pretend roles.”²³ A block of wood can be a train or telephone, taken as a symbol, and then becomes again a material thing. However this information is stored internally, the child develops powerful skills at flexibly manipulating and coordinating distinct types of representations. Mindreading in this context reflects the ordinary ways in which children become successful semioticians, expressed through abilities to understand false beliefs, compare facial expressions to emotions, guess at others’ attentive perceptions or states of mind, grasp counterfactuals, and understand concepts of appearance versus reality. The skill can be rapid, repetitive, and intricately coordinated with perceptual input and behavioral responses.

The construction of artificial languages applies just such generalized mindreading capabilities as semiotic skill. The setting is that of human computers, their perceived languages, and associated intentions regarding arithmetical tasks and symbolic manipulation. In the case of positional value, the relevant intentions involve shared perspectives on the physical symbol in terms of places, how to count them, and the process of forming higher number units. The function of positional rules is to efficiently automate the formation of number units, otherwise carried out one-by-one as a replicable piece of symbolic manipulation; that’s the redescribed pattern of the mastered ability. A cognitive model of position gets applied to the once holistic symbol pattern, now perceived as made up of constituent digit-signs taken as either used or mentioned as needed. Neither logic nor biology implies this step has to be possible for us, even given weaker but still substantial computational skills typically afforded by natural languages and additive arithmetic. The innovation comes about by directing intentional skills to language itself. Without that change, enacted by reflection on symbol usage, there’s no obvious way that facile counting skills, even basic multiplication and arithmetic, are humanly possible.²⁴

²³ (Karmiloff-Smith 1992: 132). See also (Leslie 1987, 1994) on pretend play and symbolic decoupling. As with Baron-Cohen (note 22), Leslie’s modularist framework is not critical for many basic observations.

²⁴ Chomsky made Wilhelm von Humboldt the first hero of the linguistic infinite (Chomsky 1965, v). But Johann von Herder was as visionary about the mind as Humboldt was about language, speculating that language is the product of what we today identify as reflective intentional capacities (von Herder 1772,

From this perspective, *written* numerals provide the stability and intersubjective reliability needed to establish mastery and for representational redescription to occur at all, either through grammaticalization or its relatives. Mindreading is grounded in the attention-shifting abilities needed to associate symbol with objects, behaviors, and other symbols (Baron-Cohen 1997; Tomasello 1999). With inscription, one can literally point to a digit, count its place by hand, erase it, replace it, copy it, move it, compare examples side-by-side. Vision makes it possible for stable spatial features of inscribed symbols—physical linear order, symbol concatenation, duplication, decomposition, and rearrangement—to be used in counting procedures, thereby allowing the expansion of number language and the construction of more powerful computations. To compute effectively, we need new language, like positional notation or a programming language, to express and coordinate computing goals.²⁵ Inscription not only makes symbol formation easier to describe, the descriptions can be taken as applying to the inscribed signs themselves. Thus is created a remarkable illusion of formal self-sufficiency and autonomy. In this way, core cognitive capabilities make possible considerable computational sophistication, requiring no platonic entities or other special explanations so often appearing in the philosophy of mathematics. Symbolic language itself becomes a source of metaphors for bootstrapping arithmetical skills, primarily through improved means of metalinguistic expression and interpretation. As often with metaphorical transfer, a concrete domain is used to create abstract meaning, here involving the perception of language itself, rather than the human body, features of objects, or salient experience (Deutscher 2005, chap. 4). Modern computation is as human as language and symbol usage itself, being a biologically grounded and historically evolved technology of mind and behavior.

Let's now summarize the various roles played by intentionality, symbolic perception and cognitive models in computation by an *extended Church-Turing thesis*. The ordinary Church-Turing thesis says, roughly, that all informally effectively computable functions are represented by formalized Turing machine definitions, or by any numerous equivalent definitions: Church's lambda-calculus, Post production systems, programming languages such as Fortran, Cobol, Java or C, transformational or phrase structure grammars, first-order predicate logic, many axiomatized arithmetical theories, and of course many oral arithmetics. All those equivalencies are evidence that a generic concept of effective procedure has been identified, along with the fact that no informally computable function has ever been

Footnote 24 continued

esp. 86–87). Intentionality does not start with Franz Brentano or Edmund Husserl, but has a longer history associated directly with the linked philosophies of language and mind (Aarsleff 1982).

²⁵ David Fowler writes of Simon Stevin's 1585 introduction to positional notation *De Thiende* ("The Tenth") that "Decimal fractions gave a new fluency to arithmetic which permitted, perhaps for the first time [*sic*: in Europe] the feelings that all such calculations *could now be taken for granted*, and this paved the way to the next stage, their abstraction into symbolic algebra...the people who contributed to their development—principally Stevin himself, Viète, and Harriot—were themselves calculators...*This confidence in decimal arithmetic still lies, I believe, behind our basic intuitions, even today*, underlying the real numbers, even though it is a delusion..." (Fowler 1999, 406; emphasis added). By "delusion" Fowler means that generating decimal expansions for answers to simple arithmetic operations can be very complicated, e.g. because of very slow convergence, such as occurs in many power series expansions for pi.

found that is not representable by these formalisms. But as discussed, the formation of oral arithmetic, as a grammaticalized extension of a natural language, requires special intentional skills to create the necessary representational redescrptions. So the extended Church-Turing thesis is: intentionality plays similar roles, and is needed in commensurate ways, *no matter what media* is used to express a formal system. The use of speech as a model in oral arithmetic means that intentionality doesn't disappear through inscription or alternate media, it just becomes more automated and regimented through textual and visual registers. Computation relies on the same processes needed for language use, especially language change, just realized through writing or similarly stable media to overcome memory and perceptual constraints. Intentionality plays similar roles in computation even as different media make intentional skills easier to apply. Computational complexity then increases in tandem with intended goals of enumeration, problem solutions, or provability. The extended Church-Turing thesis means that computation never completely eliminates the intentional requirements of these accomplishments, even as they become more efficiently codified and independent of any single user. "Merely formal" symbolic processing never entirely eliminates human intentionality, just as a child covering himself with a blanket can only pretend to disappear from view.²⁶

The next section concludes with a foundational perspective motivated by the role of writing in grammar-like pattern formation.

Grammaticalization in Script

We can tether the oral arithmetic thought experiment to reality through two historical observations about computation and inscription. The first involves positional grammaticalization in writing rather than speech, and hence the validity of a representational redescription model. The second, appealing to the same historical setting, involves the role inscription plays in enabling our conceptualization of discrete symbol systems, and hence modern computation, at all.

About two thousand years before the Indians, sometime around 2000 BCE, Babylonian scribal accountants developed a sexagesimal positional system combining base 60 and base 10. For example, 46 would be expressed by additively

²⁶ Wilfried Sieg and Saul Kripke have argued that the Church-Turing thesis can be *deduced* as a theorem from axioms for relevant combinatorial relations intended to represent what they take to be machine, rather than human, computations. The Church-Turing thesis, they argue, therefore does not have to be conceived as an informal thought-experiment about human computers as devised by Turing (Sieg 2008; Kripke 2000). But such theorems rely on axiom systems which are further computational idioms, expressed in artificial languages requiring apparatus for their construction as discussed in the text. Frege also overreached with his wholesale rejection of "psychologism" in mathematics (Frege 1884), though of course he lacked a modern view of how languages, including his own artificial language for predicate logic, rely on specialized cognitive and symbolic skills. John Searle has pointed out the intentional role in computation generally (Searle 1992, chap. 9), also arguing that *any* process can be identified as "computation" by intentional fiat, an observation he takes as superceding his older "Chinese Room" thought-experiment meant to demonstrate the intrinsically intentional status of language use. The text supports those views by emphasizing the particular symbolic skills used in the formation and use of artificial languages.

combining cuneiform marks for units of 1 (“O”) and another for units of 10 (“T”), as cuneiform **TTTTOOOOOO**. Larger numbers would use powers of 60, such as $1,000,000 = 4 \times 60^4 + 37 \times 60^3 + 46 \times 60^1 + 40$, reduced to symbols for **4; 37; 46; 40**, representing the number calculation via “coefficients” applied to powers of 60 marked by position. For centuries there was no zero, with unused powers left unmarked, or “skipped” by a blank, with the correct total determined by scribal context. Sexagesimal fractions for $1/60$, $1/60^2$, ... use the same notation as for 60, 60^2 , ..., with a “floating point” scale governed by accounting conventions for recording inventories or estimating quantities of food production, land, worker rations, and many other economic inputs and outputs. Positional notation was first devised to facilitate common calculations, say calculating the labor to build a wall as *wall volume* \times *bricks per unit volume* \times *man-hours per brick*. These calculations are inherently multiplicative because of the ubiquitous dimensioned units, rates, and other quantities. With cuneiform tables of constants available for typical tasks—bricks per wall volume, man-hours per bricks laid, slaves per unit field, etc.—multiplicative positional notation facilitated calculations through a cobweb of qualitative categories. The notation was also compact enough for clay tablet inscription at relatively high levels of precision, using layouts facilitating row and column sums, and visual identification of maxima, minima, or simple trends (Robson 2003).

Now, the model for Sanskrit number words was that positional value, as a multiplicative rule, was a slot-like abstraction of non- but neo-positional number words approximating the positional schema. Given that, what occurred with Babylonian script as opposed to Indian speech? The answer follows by identifying conditions for representational redescription, particularly mastery of an earlier idiom and cognitive motivations for its efficient reformulation.

Starting in the early third millennium, various non-positional systems were used to count quantities or objects of different types. There were special numeric signs, and counting or measurement conventions using different units, such as ones dedicated to calendar time; grain and cereal products; volumes of beer; dairy and milk products; worker rations; animals; implements of various types; and land areas. A typical counting system consisted of a starting unit and a finite number of multiples, each with a designated symbol: e.g. a single accounting “day” would have its sign, then this unit is counted in a “week” of 10, then three “weeks” made up 1 “month,” and 12 months a “year.” Across different administrative domains, like farming, construction, or food production, different multipliers defined higher units. So $1 \times 10 \times 6 \times 3 \times 10 \times 6$ could define a sequence of area units; then $1 \times 5 \times 6 \times 10 \times 3 \times 10 \times 6$ for some grains; $1 \times 5 \times 2$ for dairy products; and so on. These “metrological” systems only counted *numbers of fish* or *volumes of beer*, with a number count coupled to the measured item, and, as typical for a non-positional system, the highest number units fixed. There were also generic counting numbers having no definitive qualitative associations. One such system was sexagesimal with multipliers $1/2$ (or $1/10$) $\times 1 \times 10 \times 6 \times 10 \times 6 \times 10$; the counts therefore were $1/2$; 1; 10; 60; 600; 3600; 36000. Higher powers of 60, like 60^3 and 60^4 also came into use, and likewise their fractional counterparts. The powers of 60 eventually were used, as in India, to form non- but neo-positional number symbols,

probably the predecessor notations for the positional rule. By the end of the third millennium, experienced scribes and their many students dealt with multiple metrological systems, varying sequences of multiplicative factors, generic sexagesimal counting systems, and estimation or inventory tasks involving repeated multiplications across dimensioned units. Generic sexagesimal values, without qualitative categories, were initially used to translate between metrological systems, but at some time, probably stimulated by administrative dictate, it was seen that position could stand for higher number units with respect to any counting or estimation task.²⁷

Cognitively speaking, this rich practical environment of multiplicative constructs, homologous or competing number systems, and the behavioral skills to master and innovate with them, created the constructive basis for the positional innovation. The Babylonians' grammatical-like abstraction occurs in script rather than speech, and its overdetermined symbolic basis is a loosely organized collection of partially consistent, sometimes transient, systems requiring constant intertranslation. There is no single redundant format awaiting simplification, and the representational redescription differs considerably from that for Sanskrit number words. But if anything, the contrasts confirm the grammaticalization model. New sexagesimal unit formation occurs through variable position "slots" in Babylonian script, the graphic analogue of recursive compounding or affixation in speech. Cuneiform writing accelerates and facilitates symbolic accomplishments similar to those appearing in speech. According to the extended Church-Turing thesis, the intentional requirements are comparable. That completes the first observation, but we continue with its prior history.

This process of abstracting inscribed numbers also has a dramatically important precedent. The earliest number systems found in Mesopotamia counted objects, or measured amounts, for a fixed category, such as "5 goats," but no free "5" applying to other objects, like fish, bricks, or workers. The earliest counting "tokens" were three-dimensional clay shapes, such as small cones or cylinders, marked and/or sized to count some number of particular items, akin to the conjunction of number and category appearing in the later metrological systems. Use of number tokens was widespread in the Middle East for several millennia before the advent of writing around the early fourth millennium. Denise Schmandt-Besserat (1992) has argued that this numeric technology is the basis for the first writing, meaning surface inscription coordinated with general speech rather than counting. The number tokens underwent their own evolution, probably via clay envelopes, or "bullae," used for contracts. A sealed envelope would contain multiple tokens, say for numbers of fish promised and goats received. To communicate the contractual information without breaking the seal (meant to prevent falsification), the envelopes came to be marked on the outside surface with signs representing the tokens inside. This redundant system, it's further conjectured, led to the recognition that a marked surface, initially the envelope exterior, using an inscribed icon, provided the same

²⁷ Powell (1976) describes pre-Babylonian number words using multiplicative units like named powers of ten in India. On the evolution of numbers, cuneiform signs, and positional notation see (Høyrup 1994; Nissen et al. 1993). On measurement and multiplicative units see (Krantz et al. 1971, Ch. 10).

information as the original tokens, and that the inscriptions could be seen and used as independent symbols. Subsequently these new counters were replaced by separate number signs which could freely combine with signs for grain, workers, and so forth. These are the earliest “pure” numbers, created as yet another representational redescription involving multiple media and cross-referenced sign systems, and anticipating the problems of “coupled” units in much later and more complicated metrological systems. For Schmandt-Besserat, the earlier separation of number and category was the birth of pictographic writing to communicate generally, because the “category” half of, say, a fused number sign “5 goats” could be taken as “goats.” The writing concept is implied by a conceptual model which allows either the number or category term to be treated as a variable “slot”, such as ___ *goats* or 5 ___ or ___ ___, hence involving similar intentional processes occurring in grammatical constructions.

Whether this symbolic evolution is the first instance of writing or not, these early inscriptions and number formations rely on the powerful decoupling skills we apply to symbol formation in diverse media.²⁸ Later still, in Mesopotamia, the Middle East, and around the Mediterranean, elemental pictographs were taken as sounds, like a picture of an eye taken as the sound for “I”, thereby allowing inscription to become a simple model for speech. After that first “rebus”-style writing, pictures were discarded by some as irrelevant, and icons radically evolved into streamlined graphic stereotypes directly representing syllabic speech, thus taking speech modeling a step further. Then later, syllabic sounds were analyzed into individual phonemes using alphabetic renderings composed of consonant, or consonant and vowel, signs. Sometimes considered a culmination of alphabetic technology, the Greek alphabet probably arose because Greek needed vowels to differentiate words (like *bed* vs. *bad*), while Semitic languages used vowels only for inflections and lexical information carried completely by consonants: so in Arabic *ktb/write* can inflect as *katab/he wrote* or *katabi/I wrote*. Perhaps because it was easier to leave vowels unmarked, and determine them by context, some Semitic writing systems did not use them. But in being transferred to Greek, too many words could not be modeled, so an alphabetic system using consonants and vowels, much like ones used today, was devised to make vowels easier to handle in the target language. The procession of writing systems in the ancient world are historic examples of cognitive technologies created through culturally transmitted and refined representational redescription. And begun, apparently, with number tokens and their “fused” conceptual elements.

All this confirms the roles of cognitive models of symbolic form and representational redescription in arithmetic, whether inscribed or not. But a new twist has appeared too. Historical writing systems are devised via some linguistic model, created through the design of the system itself, such as needed to perceive whole words, or syllables, or phonemes, or some combination of these, while

²⁸ “The ‘great invention’ [of writing] was almost certainly the prehistoric move from a token-iterative to an ‘emblem-slotting’ [a generic number sign combined with a separate category] for recording numerical information...Slotting is a structural technique we now regard as intrinsic to language; and nowhere more typically than in the way languages deal with counting.” (Harris 1986, 145). On slotting see also (Tomassello 2003, 122–126).

ignoring, e.g., intonation, stress, melodic contours, or communicative intent (Olson 1994; Coulmas 2003). By such modeling, inscribed signs catalyze relevant memory, and that success motivates us to see language in terms of whole word meanings, syllabic parts, morphemes or phonemes. The writing system induces theory-laden perceptions of the language to which it is applied, making it in turn amenable to representation in script. In the case of the Greeks, for example, they saw that Phoenician script was a poor model for Greek, so they added a set of vowels separated from the consonants to make it work. But the new linguistic model still leaves out many features of speech, and has no representation independently of the new Greek alphabet. Consequently writing systems create a fuzzy, many-many relationship between speech and text. Therefore it cannot be *prima facie* assumed that the implicit linguistic models, however useful, are just copies of speech, or vice versa. Writing systems rely for their use on our taking a certain view of language, and once internalized, it is difficult not to experience those somewhat magical effects.

Modern mathematics and logic take for granted these unusual powers afforded by writing whenever a “foundation” assumes notions such as discrete symbols, lists of symbols, their concatenation and organization into functional categories, and strict rules automating the production of useful, and typically potentially infinite, sets of proofs, theorems, or other computations. We freely assume that a formal system can be realized in any media we like, and the nature of mathematical abstraction means there’s nothing special about writing. But it is difficult to provide examples of formalisms which don’t rely on the concrete, and especially spatial, expression of writing to define formal, computational, or logical concepts.

Hence it is tempting to think of natural language and its allied usage skills as a new computational foundation. As shown, we rely on similar intentional capabilities for much modern arithmetic and mathematics. The oral arithmetic thought experiment gets us constructively from natural language, conceived in terms of discrete phonemes, stems, and roots, through a natural object-language, to artificial metalanguages sufficient for generative grammar and universal computation. From this perspective, phonemes are the paradigm of a discrete symbol as assumed for mathematics; they define a physio-mechanical isomorphism, or duality, between the *phonetic* sound form and a *phonemic* function as differentiating symbol. Certainly modern mathematics takes such entities for granted, like physical points, lines, or surfaces in geometry.²⁹ The linguistic principle of *duality of patterning* (or “double articulation”) is that phonemes are jointly “mere marks”—reductively defined via matrices of *distinctive features* characterizing the flow of air through the vocal apparatus—and functional signs. They are physical events *servicing* as symbolic atoms: e.g. the /s/ in *cats* is both “just” a sound *and* a marker of plural function; or /g/ and /d/ are “just” sounds but differentiate *god* from *dog*. Duality of patterning is the naturalized counterpart to an unstated principle, ubiquitous in axiomatic and computing thought, for creating useful symbolic forms nearly at will.

Unfortunately for this line of thought, duality of patterning, as a formal and mathematical working assumption, cannot be discharged via natural language. The

²⁹ See note 12 above and text.

reason is that phonemes, as discrete forms, have generally *not* been found to have a physical basis (Crystal 1987, 145). In speech, phonemes are constantly under adjustment, conforming to requirements of adjacent sounds, making speech a quasi-parallel and continuous process through which we unconsciously optimize, for example, tongue position or needed breath for the upcoming sound. So much so that, as put by Steven Pinker, boundaries between words, which would apparently be identifiable through a phoneme's distinctive features, are effectively "hallucinated." Word and phonemic boundaries are generally imposed as a perceptual gestalt on an otherwise continuous speech stream, a product of our considerable powers of categorical perception (Pinker 1994, 159). The foundational role of discrete symbols, if appealing to phonemes, abruptly returns us to their perception.³⁰ And as far as we can describe those perceptions in discrete terms, we rely on models motivated by descriptions enabled by writing systems, as just mentioned. If we *assume* some discrete starting point, as in computation and logic, and as was done by Pāṇini, and after him many modern linguists, then we can construct computation using only natural language and intentional skills as described through the oral arithmetic thought experiment. But that "inductive basis" for a thoroughly naturalized understanding of where computation and much of modern mathematics comes from, itself has a dual nature combining form and function, as do higher-order syntactical constructions.³¹ Thus the probable hypothesis that Pāṇini's fabulous oral grammar itself has a *written* foundation through its discrete phoneme set, the *Śivasūtras* (Goody 1987, chap. 4), here is exposed as a deep and remarkably successful assumption for artificial language creation. In this way, mathematics relies on a folk model of symbols and language, much as pre-relativistic physics assumed a Newtonian commonsense framework of absolute space and time.

Does this matter for mathematics? Script facilitates *virtual platonism*, meaning the useful and reproducible illusion that number concepts and systems are independent of their historical origins and their reliance on intentional skills. Virtual platonism, not real platonism, is all that is needed for mathematics and symbolic computation. Arthur C. Clarke, late creator of *2001: A Space Odyssey*, wrote that "any sufficiently advanced technology is indistinguishable from magic." Which means there is no magic, just human skills and products so intricately assembled and deployed to create the perfect illusions of symbolic algorithms: first by Babylonian scribes, then Indian grammarians and mathematicians, and now modern mathematicians, logicians, computer scientists, and linguists. Inscription combined with symbolic intentionality together make it possible for us to bootstrap the most elementary counting procedures, starting with notched sticks and a few number words, into advanced computations of all kinds.

But that exquisite construction depends on a sleight-of-hand. With an inscribed alphabet used to model speech, we can separate what is seen from what is heard, and discern the differences between source and target. But with number symbols the

³⁰ Stimulated by Donald Knuth's TeX typesetting language, (Hofstadter 1985) argues that no fundamental criteria characterize variations across all font designs, say for the letter "a". Thus graphemes are not better defined than phonemes (Coulmas 2003, 204).

³¹ See (Jakobson 1990, 240) on this synthetic duality in phonemes, and (Tomasello 2003, chap. 5) and references there on similar form-function dependencies in syntactical analysis.

situation changes. For counting, inscription naturally becomes its own model. The idealized characteristics we give to writing, enabling a logic of symbols, are taken up without fanfare into the idea of a discrete formal system. Then we turn around and use inscription as an intended realization, as in written proofs or computations. That is how a mathematical idealization relies on a folk model of inscribed signs. The situation is analogous to estimating parameters using the very dataset to be forecast. Metamathematical thought is possible because mathematics and computation are so thoroughly metamathematical to begin with, just as writing systems are metalinguistic. To be sure, it's all pragmatically useful, but symbolic logic, in a root sense, relies on this blur of hands as viewed by a willful audience of symbol users. Mathematical language, thought of as formalized symbol processing, stands to *itself* in the same way that alphabets and other inscription systems stand to natural language. It is its own double.

As put by Ferdinand de Saussure—actually perhaps a student at Saussure's course in general linguistics, an irony of speech recorded as text if there ever was one—“Language can also be compared to a sheet of paper: thought is the front and sound the back; we cannot cut one side without also cutting the other. So also in language, sound cannot be isolated from thought, or thought from sound”(Saussure 1915, 111). Computation similarly is a many-sided symbolic technology of mind, media, and behavior. Its sheets, as we choose to find them, are sided with material inscriptions and sounds; discrete symbols and symbolic patterns of many kinds; calculations, computations and proofs; and descriptions of all these, their changes, and transformations. On these pages are found the numbers and calculations of modern mathematics and computation.

References

- Aarsleff, H. (1982). *From Locke to Saussure: Essays on the study of language and intellectual history*. London: Athlone.
- Barber, C. (2000). *The English language* (Rev. ed.). New York: Cambridge University Press.
- Baron-Cohen, S. (1997). *Mindblindness: An essay on autism and theory of mind*. New York: Cambridge University Press.
- Bloom, P. (1994). Generativity within language and other cognitive domains. *Cognition*, 51, 177–189. doi:10.1016/0010-0277(94)90014-0.
- Bloomfield, L. (1933). *Language*. New York: Holt.
- Boas, F., & Powell, J.W. (1911/1966). *Introduction to handbook of American Indian languages*. Lincoln: University of Nebraska Press.
- Boolos, G., & Jeffrey, R. (1979). *Computability and logic*. New York: Cambridge University Press.
- Bybee, J. (2003). Cognitive processes in grammaticalization. In M. Tomasello (Ed.), *The New psychology of language vol. 2: Cognitive and functional approaches to language structure* (pp. 145–168). Mahwah, New Jersey: Lawrence Erlbaum.
- Chomsky, N. (1963). Formal properties of grammars. In R. D. Luce, et al. (Eds.), *Handbook of mathematical psychology* (Vol. II, pp. 323–418). New York: Wiley.
- Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT.
- Chomsky, N. (1980). *Rules and representations*. New York: Columbia University Press.
- Clark, H. (1996). *Using language*. New York: Cambridge University Press.
- Coulmas, F. (2003). *Writing systems*. New York: Cambridge University Press.
- Crystal, D. (1987). *The Cambridge encyclopedia of language*. New York: Cambridge University Press.
- Datta, B., & Singh, A. N. (1935/2001). *History of Hindu mathematics I, II*. Delhi: Bharatiya Kala Prakashan.

- Davis, M. (Ed.). (1965). *The undecidable: Basic papers on undecidable propositions, unsolvable problems and computable functions*. Hewlett, New York: Raven Press.
- Dehaene, S. (1997). *The number sense: How the mind creates mathematics*. New York: Oxford University Press.
- Deutscher, G. (2005). *The unfolding of language*. New York: Metropolitan Books.
- Dijksterhuis, E. J. (1938/1987). *Archimedes* (C. Dikshoorn, Trans.). Princeton, New Jersey: Princeton University Press.
- Everett, D. (2005). Cultural constraints on grammar and cognition in Pirahã. *Current Anthropology*, 46, 621–646. doi:10.1086/431525.
- Fowler, D. H. (1994). Could the Greeks have used mathematical induction? Did they use it? *Physis*, 28, 273–289.
- Fowler, D. H. (1999). *The mathematics of Plato's academy* (2nd ed.). Oxford: Clarendon.
- Frege, G. (1884/1968). *The foundations of arithmetic* (J. Austin, Trans.). Evanston, Illinois: Northwestern University Press.
- Gentner, D., & Goldin-Meadow, S. (Eds.). (2003). *Language in mind: Advances in the study of language and thought*. Cambridge, MA: MIT.
- Gillon, B. (2007). Pāṇini's Aṣṭādhyāyī and linguistic theory. *Journal of Indian Philosophy*, 35, 445–468. doi:10.1007/s10781-007-9027-3.
- Goody, J. (1987). *The interface between the written and the oral*. New York: Cambridge University Press.
- Harris, R. (1986). *The origin of writing*. La Salle, Illinois: Open Court.
- Hofstadter, D. (1985). Metafont, metamathematics, and metaphysics. In *Metamagical themes: Questing for the essence of mind and pattern* (pp. 260–300). New York: Basic Books.
- Høyrup, J. (1994). Mathematics and early state formation, or the Janus face of early Mesopotamian mathematics: Bureaucratic tool and expression of scribal professional autonomy. In *In Measure, number, and weight* (pp. 45–88). Albany, NY: State University Press of New York.
- Ifrah, G. (2000). *The universal history of numbers: From prehistory to the invention of the computer* (D. Bellos et al., Trans.). New York: Wiley.
- Jackendoff, R. (2002). *Foundations of language*. New York: Oxford University Press.
- Jakobson, R. (1990). *On language*. L. Waugh & M. Monville-Burston (Eds.), Cambridge, MA: Harvard University Press.
- Kadvany, J. (2007). Positional value and linguistic recursion. *Journal of Indian Philosophy*, 35, 487–520. doi:10.1007/s10781-007-9025-5.
- Karmiloff-Smith, A. (1992). *Beyond modularity: A developmental approach to cognitive science*. Cambridge, MA: MIT.
- Krantz, D., Luce, R. D., Suppes, P., & Tversky, A. (1971). *Foundations of measurement volume 1: Additive and polynomial representations*. New York: Academic Press.
- Kripke, S. (2000). From the Church-Turing thesis to the first-order algorithm theorem. In *Proceedings of the 15th annual IEEE symposium on logic in computer science* 177. Washington, DC: IEEE Computer Society.
- Leslie, A. (1987). Pretense and representation: The origins of “theory of mind”. *Psychological Review*, 94, 412–426. doi:10.1037/0033-295X.94.4.412.
- Leslie, A. (1994). Pretending and believing: Issues in the theory of ToMM. *Cognition*, 50, 211–238. doi:10.1016/0010-0277(94)90029-9.
- Levinson, S. (2003). *Space in language and cognition: Explorations in cognitive diversity*. New York: Cambridge University Press.
- McWhorter, J. (2001). *The power of Babel*. New York: Henry Holt.
- Minsky, M. (1967). *Computation: finite and infinite machines*. New York: Prentice-Hall.
- Nissen, H., Damerow, P., & Englund, R. (1993). *Archaic bookkeeping: Writing and techniques of economic administration in the ancient near east* (P. Larsen, Trans.). Chicago: University of Chicago Press.
- Olson, D. (1994). *The world on paper*. New York: Cambridge University Press.
- Ostler, N. (2005). *Empires of the word*. New York: Harper Collins.
- Pinker, S. (1994). *The language instinct*. New York: William Morrow.
- Powell, M. (1976). The antecedents of old Babylonian place notation and the early history of Babylonian mathematics. *Historia Mathematica*, 3, 417–439. doi:10.1016/0315-0860(76)90071-9.
- Pullum, G., & Scholz, B. (2005). Contrasting applications of logic in natural language syntactic description. In ed. P. Hájek et al. (Eds.), *Logic, methodology and philosophy of science: Proceedings of the twelfth international congress* (pp. 481–503). London: King's College Publications.

- Renou, L. (1941/1972). Les connexions entre le rituel et la grammaire en sanskrit. In J. F. Staal (Ed.), *A reader in the Sanskrit grammarians* (pp. 435–469). Cambridge, MA: MIT.
- Robson, E. (2003). Tables and tabular formatting in Sumer, Babylonia, and Assyria, 2500 BCE–50 CE. In M. Campbell-Kelly, et al. (Eds.), *The history of mathematical tables from Sumer to spreadsheets* (pp. 18–47). New York: Oxford University Press.
- Sapir, E. (1921/1949). *Language: An introduction to the study of speech*. San Francisco: Harcourt Brace.
- Saussure, F. (1915/1959). *Course in general linguistics* (W. Baskin, Trans.). C. Bally et al. (Eds.), New York: McGraw-Hill.
- Schmandt-Besserat, D. (1992). *From counting to cuneiform* (Vol. 1). Austin: University of Texas Press.
- Searle, J. (1983). *Intentionality: An essay in the philosophy of mind*. New York: Cambridge University Press.
- Searle, J. (1992). *The rediscovery of the mind*. New York: Cambridge University Press.
- Sharma, R. N. (1987). *The Aṣṭādhyāyī of Pāṇini I: Introduction to the Aṣṭādhyāyī as a grammatical device*. New Delhi: Munshiram Manoharlal.
- Sieg, W. (2008). Church without dogma: Axioms for computability. In B. Lowe, et al. (Eds.), *New computational paradigms* (pp. 139–152). New York: Springer.
- Slobin, D. (2003). *Language and thought online: Cognitive consequences of linguistic relativity*. In Gentner and Goldin-Meadow (pp. 157–192).
- Staal, J. F. (1975). The concept of metalanguage and its Indian background. *Journal of Indian Philosophy*, 3, 315–354. doi:10.1007/BF02629150.
- Staal, J. F. (1988). Euclid and Pāṇini. In *Universals: Studies in Indian logic and linguistics* (pp. 143–160). Chicago: University of Chicago Press.
- Staal, J. F. (1990). *Ritual and mantras: Rules without meaning*. Delhi: Motilal Banarsidass.
- Staal, J. F. (1995). The Sanskrit of science. *Journal of Indian Philosophy*, 23, 73–127. doi:10.1007/BF01062067.
- Staal, J. F. (2006). Artificial languages across sciences and civilizations. *Journal of Indian Philosophy*, 34, 87–139.
- Tomasello, M. (1999). *The social origins of human cognition*. Cambridge, MA: Harvard University Press.
- Tomasello, M. (2003). *Constructing a language: A usage-based theory of language acquisition*. Cambridge, MA: Harvard University Press.
- Unguru, S. (1991). Greek mathematics and mathematical induction. *Physis*, 28, 273–289.
- von Herder, J. G. (1772/2002). Treatise on the origin of language. In *Philosophical writings* (M. Forster, Ed. and Trans.) (pp. 65–166). New York: Cambridge University Press.
- Whorf, B. L. (1956). J. Carroll (Ed.), *Language, thought and reality: Selected writings of Benjamin Lee Whorf*. Cambridge, MA: MIT.
- Williams, M. (1846/2005). *An elementary grammar of the Sanscrit language*. New Delhi: Cosmo.
- Woepcke, F. (1863). Mémoire sur la Propagation des Chiffres Indiens. *Journal Asiatique, Sixth Series*, 1, 442–529. (Reprinted in F. Sezgin (Ed.). *Franz Woepcke, Études sur les Mathématiques Arabo-Islamiques*, vol. 2. Frankfurt: Institute for the History of Arabic-Islamic Science and the Goethe University, 1986).