Techset Composition India (P) Ltd., Bangalore and Chennai, India THPL1121439.TeX Page#: 22 Printed: 29/12/2015

HISTORY AND PHILOSOPHY OF LOGIC, 2015
http://dx.doi.org/10.1080/01445340.2015.1121439

Taylor & Francis
Taylor & Francis Group

Pāṇini's Grammar and Modern Computation

JOHN KADVANY
Policy & Decision Science, USA
john@johnkadvany.com

Received 13 August 2015 Accepted 14 November 2015

Pāṇini's fourth (?) century BC Sanskrit grammar uses rewrite rules utilizing an explicit formal language defined through a semi-formal metalanguage. The grammar is generative, meaning that it is capable of expressing a potential infinity of well-formed Sanskrit sentences starting from a finite symbolic inventory. The grammar's operational rules involve extensive use of auxiliary markers, in the form of Sanskrit phonemes, to control grammatical derivations. Pāṇini's rules often utilize a generic context-sensitive format to identify terms used in replacement, modification or deletion operations. The context-sensitive rule format is itself defined using Pāṇini's more general method of auxiliary markers, the latter used to define many dozens of linguistic categories and rules controlling derivations of Sanskrit sentences through the manipulation of 'non-terminal' and 'terminal' symbols. This technique for controlling formal derivations was rediscovered by Emil Post in the 1920s and later shown by him to be capable of representing universal computation. The same implicit computational strength of Pāṇini's formalism follows as a consequence: while Pāṇini's Sanskrit grammar is computationally limited, the metalanguage through which his formalism is defined can be directly used to define any rule-based system by mimicking standard formal language definitions as an extension of the grammatical system proper. Pāṇini's formal achievement is historically distinctive, as derivations of grammatically correct, spoken Sanskrit, are designed for oral recitation, with the grammar itself constructed as an organic extension of the spoken object language. Pāṇini's formulation of what amounts to an orally realized symbolic calculus stands in contrast to the implicit inscriptional methods of contemporary formalisms, such as Gottlob Frege's appropriately named *Begriffsschrift* and the early computing paradigms of Post and Alan Turing. Nonetheless, contemporary views on the cognitive status of phonemic recognition and historical writing systems support the conjecture that, in spite of Pāṇini's rigorous oral formulation, construction of the grammar almost surely relied on alphabetic writing. Q1

1. Grammar and computation

For purposes of this paper, 'computing language' means a formal calculus capable of representing universal computation according to the rules of some formal language, explicitly described through a metalanguage characterizing language categories and expression formation. The language should also have some implicit or explicit realization in some media, such as inscription or electronic storage. In this sense, modern machine and high-level programing languages are computing languages. So too are the classical computing models of Emil Post, Alan Turing, Alonzo Church, Stephen Kleene and others, including Kurt Gödel's formalization of metamathematics via number theory. Though not 'programing' languages intended for machine implementation, the classical models are all 'computing languages' by virtue of an operational formalism which can be then used to represent all effective procedures. Gottlob Frege's first-order logic may be included here just because, as recognized by Church and Turing, Gödel's number-theoretic coding, as well as their own formalisms, may be translated into axioms expressed in first-order logic (and so showing the valid sentences of first-order logic to be undecidable). We tend to think of formalisms capable of expressing arbitrary algorithms as thoroughly modern, typically late nineteenth and early twentieth century creations. It is also a modern idea, following Gödel, to see how to describe the derivational rules of a formal language also through the language, so that object- and metalanguage are unified as one.

CE: KS QA: Coll:

But the nineteenth and early twentieth century formalisms for algorithmic expression are not the earliest such, by about two millennia. The first computing language – again, for our purposes, a generic formalism, described through a metalanguage for representing exact generative symbolic procedures of any kind – was devised circa 350 BC by the Indian grammarian and linguist Pāṇini.[1] The formalism is not identical with Pāṇini's Sanskrit grammar, but is a significant part of it, constituting the grammar's formidable derivational methods. Those methods are allied with several centuries of Indian linguistic theory to define the grammar as a whole. It has been linguistic folklore for decades that Pāṇini was formally sophisticated, and adept at defining linguistic categories and rules for their application (*Ostler* 2005, p. 181). At the start of *Aspects of the Theory of Syntax*, Noam Chomsky cites Pāṇini's grammar as a generative ancestor (*Chomsky* 1965, p. v). As put by the Sanskritist S.D. Joshi, the grammar 'is also a generative calculus, which is actually [its] main thrust' (*Joshi* 2009, p. 5). Pāṇini's formal method has been compared specifically with the Backus-Naur style rules of programing language definition (*Ingerman* 1967).[2]

Left open is the scope of Pāṇini's formal methods, in contrast to his linguistics, and their relationship to modern computing concepts. This paper completes the characterization of Pāṇini's formal skills by noting that his grammar relies on the method of auxiliary markers, or terminal and non-terminal symbols, as the primary heuristic for expressing formal rules. Pāṇini's rules are defined using a semi-formal metalanguage for defining linguistic categories and rules which operate on those categories to generate Sanskrit expressions.

Pāṇini's basic method was rediscovered, we can now say, in the 1920s and 1930s by Emil Post through his production/rewrite systems. As an example of rewrite methods, three-letter palindromes can be enumerated using a non-terminal marker p and terminal symbols a, b, c. Expressions start with pa, pb, pc and p. Three rewrite rules allow an expression pX, starting with p, to be rewritten as $paXa$, $pbXb$ or $pcXc$. Palindromes using a, b, c are generated and marked with an initial p. A final rule allows any expression pX to drop the p (or substitute a null sign), leaving only terminal symbols. Alternatively, in Backus-Naur form, the same expressions could be defined recursively as $P := blank \mid a \mid b \mid c \mid aPa \mid bPb \mid cPc$, with P a metalanguage label for object language palindromes, and *blank* standing for an empty or null string. Post's methods are now widely used as a formalism for defining programing languages and the programs written in them. In this way, Post's method become a standardized metalanguage for language definition and program validity.[3]

Post then also *proved*, as Pāṇini could not even conceive, that his systems were capable of universal computation. But then that fact has also to be true of Pāṇini's grammar, even as the latter is meant for computationally modest linguistic derivations and not calculation nor computation generally, though the latter are wholly possible. Parallel to Euclid's codification of the earliest deductive systems, including proof by contradiction, Pāṇini's Sanskrit grammar formulates and applies the world's first formal language for generic symbolic manipulation. As put by the late Frits Staal, Pāṇini is indeed the 'Indian Euclid' (*Staal* 1965a).

[1] *Houben* 2009 (p. 6) suggests that Pāṇini's use of *rūpya* at sūtra 5.2.120 refers to a coin appearing only from the fourth century.

[2] John Backus, lead designer of Fortran, said he adapted Post's methods to describe an early version of Algol (*Backus* 1959, p. 129; 1980, p. 132). Pāṇini preceded Post, so Ingeman's partial comparison is apt. See also note 14.

[3] In the same way, first-order logic is a metalanguage for theories expressed in predicate logic; Turing machines, Church's lambda calculus and Kleene's partial recursive functions do the same for their respective computing idioms. As explained below, Pāṇini solves the problem of bootstrapping a metalanguage from its targeted object language by utilizing affixing methods of Sanskrit itself. On Post systems, see *Davis et al.* 1994, ch. 5, *Minsky* 1967, ch. 12; for Post's history including work antedating Turing, see *Urquhart* 2009, sec. 4.

Pāṇini's grammar has the further property, relevant to contemporary programing languages, that it is formulated for *oral* recitation, not inscription; indeed, Pāṇini's formalism can be construed as a grammatical generalization of the spoken Sanskrit object language which the grammar describes. The grammar's derivations are designed to produce well-formed Sanskrit *speech*, using a finite set of discrete Sanskrit phonemes as its elementary symbols *Saṁskṛta*, prefixing the root *kṛ* with *saṁs*, itself means 'polished, well-formed'. In this way, Pāṇini's grammar is the realization of a computing environment as formally recited human speech. That is consistent with the modern idea that computing software can be expressed in any medium compatible with the representation of discrete symbols and their systematic manipulation. Whether the grammar may also have been *formulated* lacking inscriptional help, especially that of alphabetic writing, is a separate question taken up in paper's final section, along with consequences for modern computing languages and formal systems.

What follows is an expository summary of Pāṇini's grammar including the metalinguistic apparatus through which its formalism is defined. The main goal is to show Pāṇini's expertise at utilizing Post-style rewrite throughout the construction and operation of his grammar. By way of historical context, Pāṇini's grammar is motivated to construct 'certificates of authenticity', so to speak, for Sanskrit expressions, for both scientific and ideological reasons. Procedural exactness has deep roots in Indian culture, particularly via older traditions of ritual theory. The earliest Indian linguistic theories were conceived through the latter, including the characterization of grammar as representing continuous speech (*saṃhitā*) using artificial discrete simplifications (*pada*). More generally, language and linguistics had a preeminent scientific role in ancient India, comparable to geometry and astronomy in Greece, but with a complementary prestige associated in India with algorithmic thinking of all kinds. The oldest theoretical formulations of the topic appear to be those of so-called ritual 'manuals', guiding explicit ritual design and execution in the *Vedas* and elsewhere. Already here are several grammatical ideas, including *sūtras* as rules, and rule guidelines, or *paribhāṣās*, describing ritual protocols and their execution. Ritual procedures were seen as recursive in that one ritual could be designated to precede, follow, or be embedded as a complete step within another, and with such steps repeatable.[4] These early concepts were considerably extended by Pāṇini for Sanskrit linguistics, created originally as a subtopic of ancient Indian ritual analysis (See *Renou 1941*; *Staal 1990*).[5]

2. Pāṇini grammar

Like many modern grammars, and much like all modern formal languages, Pāṇini's grammar includes a tiered hierarchy of progressively more powerful representations, the 'levels' being: physical sounds to symbolic phonemes; phonemes to meaningful

[4] As an example of recursive procedural embedding: (i) I enter the room. I leave the room. (ii) I enter the room. *I turn on the TV. I turn off the TV*. I leave the room; etc. In ritual the activities might include oblations, chants, participant actions, etc. (*Staal 1990*, Part II). Patañjali (ca. second century BC) compared the grammatical infinity of language to the same feature in ritual theory: 'There are indeed linguistic expressions which are never used … Even though they are not used, they have of necessity to be laid down by rules, just like protracted *sattras*,' a type of Vedic ritual performed only by priests (*Staal 1990*, p. 89). Pāṇini, Kātyāyana and Patañjali are considered the three great linguists of ancient India. Modern expressions of generativity famously appear in Antoine Arnauld and Claude Lancelot's 1660 *Port-Royal Grammar* and Wilhelm von Humboldt (*von Humboldt 1999/1836*, p. 91), for whom *thought* is potentially infinite, hence requiring the same for language.

[5] For additional features of ritual theory specialized for Indian linguistics see note 10. On Sanskrit as the Indian language of science, see *Staal 1995*. On language as goddess and its social aspect see *Rig Veda* hymns 10.71, 10.71.2, 10.125 and *Staal 2008*, p. 291. For history of Sanskrit and its influence, see *Ostler 2005* (ch. 5) and *Pollock 2006*.

morphemes; and morphemes to syntactically well-formed words and sentences.[6] The grammar includes a great deal of implicit semantics through its linguistic content and a set of basic semantical categories used to initiate derivations, as explained below. Pāṇini's finitary basis includes the basic set of Sanskrit phonemes (*Śivasūtras*); Sanskrit verbal roots (*Dhātupāṭṭha*) and nominal stems (*Gaṇapāṭha*), from which words and compound words are formed[7]; and, for metalinguistic purposes, additional phonemic markers, used as affixes, to control the derivation of Sanskrit words and sentences. Pāṇini's grammar is known as the *Aṣṭādhyāyī*, meaning 'eight books', with some 4000 rules codified as terse mnemonic sūtras (literally 'threads'), conventionally numbered *b.c.n* for book *b*, chapter *c*, sūtra *n*.[8]

Along with the initial symbol sets, categorical definitions are introduced which are given functional roles through the grammar. Terms such as *vṛddhi* (sound segment), *dhātu* (nominal stem), *pada* (fully derived word) and about a hundred others are of this type, providing a stock of linguistic concepts which the grammar's rules organize and act upon. Definitions may include simple lists, a category of words, a category of words labeled in a certain way, or even a collection of rules appearing at a given location in the grammar. Such definitions occur through the grammar's metalanguage, with *saṃjñā* referring broadly to many types and subcategories of definitions. Technical terms may be ordinary Sanskrit or specially invented terms; the latter are used for 'theoretical' grammatical concepts and categories, and the former used for non-grammatical 'givens'. For example, phonetic *sounds* are taken as given and 'primitive', but not phonological classifications of word segments. Metagrammatical terms are also taken as given, being assumed as common ground, or 'basic equipment', needed for use of the grammatical system (*Kiparsky* 1980, ch. 6). *Saṃjñin* is the 'object' to which a term is assigned, such as a list of vowels, a type of compound word, words with a fixed set of assigned affixes and so forth. This widely applied apparatus for defining symbolic categories implicitly makes the grammar formally general, since most any symbolic category can be so defined from the starting symbol set. Pāṇini's definitions transition from informal linguistic notions to their formal characterization in the grammar. 'Derivation' can be taken not entirely, but very much, in the modern sense, as rule-governed, step-wise expression formation. The typical action or event is to rewrite – or 'respeak' – a current expression E with some modified E'. Important differences between Pāṇini's and typical modern formal derivational steps are indicated below.

The user of the grammar, like the user of a formal proof system or programing language, will start with some Sanskrit target word, compound word or sentence in mind as the goal. Generally, Pāṇini's grammar is 'a derivational word-generating device' (*Joshi*

[6] No theory of language accompanies the grammar, hence this starting point is justified only by contemporary interests and linguistic theory; on emergent features of Pāṇini's grammar, such as the tiered structure noted in the text, see *Kiparsky* 2009 (p. 34).

[7] For languages not relying (like English) on word order and serial prepositions (e.g. *at*, *by*, *in*, *from*, etc.), generativity occurs through other means, while Sanskrit allows several types of recursively defined compound words. The simplest are those converting a list like *horse*, *man*, *stream*, *sun* into a single word which in English would be combined using 'and', called a *dvandva* compound by linguists still today. Other compounds modify a single member (either the first or last constituent), or combine words using a shared case structure (*husband of the sister of Sally*). Newly created compound verbs or nominals can then be inputs to appropriate syntactic slots, leading directly to ordinary recursive constructions. Sanskrit poets may create compounds using more than a dozen constituents, although historically such usage may have appeared only after Pāṇini's grammar. On the centrality of Sanskrit compounds, see *Williams* 1846 (ch. 9) and *Staal* 1995 (p. 84) on complex recursive branching. An important example of Sanskrit compounding is *positional* Sanskrit number words, which can be construed as a grammaticalization of numeric place or position (*Kadvany* 2007, p. 501).

[8] On Pāṇini's grammar generally, see *Cardona* 1988, *Sharma* 1987; and from more specific linguistic perspectives, see *Gillon* 2007 and *Staal* 1988 (Part II). *Mishra* 2009 and *Scharf* 2009 address the grammar's structure in the context of Sanskrit computational linguistics.

2009, p. 2). The grammar is used constructively, like a spreadsheet, and is *not* generally capable of directly 'testing' candidate expressions for grammaticality, though invalid derivations will at some point fail. In the end there is no master definition of 'grammatical expression', no 'if and only if' statement. Rather, an expression qualifies as grammatical just in case it can be produced by the rules. Pāṇini's rules are also 'non-deterministic', meaning derivation options are sometimes possible, such as putting a sentence into an active or passive voice. The non-deterministic formulation further compacts the grammar and reduces its size for memorization. As a means of better characterizing empirical speech, Pāṇini also marks variant usage as occurring 'usually' versus 'rarely', or 'sometimes' (*Kiparsky 1980*, ch. 1). Generally, any applicable rule is applied to any derived form until no more rules are applicable, subject to constraints, not all of which are stated explicitly, preventing incorrect derivations. As a modern parallel, interpretive rules written in the programing language Mathematica are similarly applied: user input is scanned and rewritten until that is no longer possible, including detection of an error condition. As in most empirical linguistic analysis, there are sentences whose syntax is not quite produced by the grammar, or produced only through some ad hoc interpretation of rule application. Nonetheless, the grammar is recognized as one of the greatest ever devised for its many linguistic insights, ingenious methods, comprehensiveness and rigor (*Bloomfield 1934*, p. 11).

Needed roots and stems are user-selected to initiate the derivational process,[9] and because Sanskrit is mostly a free-order language, like Latin or Old English, the ordering of these elements is largely irrelevant (through ordering *within* several types of compounds can matter). From this starting point, metalinguistic rules (*paribhāṣās* – a term assumed by the tradition but not used by Pāṇini) are used to mark roots and stems as having their intended syntactic roles, using six functional categories which today's linguists may characterize as agent, goal, patient, instrument, location and source. As put by Paul Kiparsky, 'Pāṇini's grammar represents a sentence as a little drama consisting of an action with different participants, which are classified into role types call *kārakas* [which are] roles, or functions assigned to nominal expressions in relation to a verbal root' (*Kiparsky 2005*, p. 60). While such choices are made by the user, the *kāraka* metarules list the categories and rules for using them. 'Pāṇini thus takes meanings into consideration from the very outset of a derivation' (*Cardona 1988*, p. 160). Because of that, and the need to interpret rules through working knowledge of Sanskrit, the grammar is not sharply divided into phonology, morphology syntax and semantics as in some modern linguistics (*Cardona 2009*, p. 14).

From this starting point of the *kāraka* roles and selected proto-words, Pāṇini's metalanguage guides the arduous process of identifying and applying relevant operational rules (*vidhi*) which step-wise transform roots and stems into valid Sanskrit words and sentences, primarily through affixing and compounding. The proper prioritization, exception-allowing, rule-blocking and other uses of operational rules are also laid out by the guiding metarules. The process is comparable to the formation of an individual, concrete and well-formed program by the rules of the programing language in which it is expressed. The 'output' here is a single well-formed word, or set of words constituting a sentence. All through the process, rule application relies on considerable expertise, and some subjective judgment, for rule identification and application. While employing a rigorous formalism throughout, the organization of rules and their application is subtle and intricate, as happens with the linguistic analysis of many natural languages. This blending of linguistic theory,

[9] This approach for initiating derivations is a received account of how the grammar is to be applied, e.g. the grammar 'clearly requires a user who wants to check and possibly improve a preliminary statement' (*Houben 2009*, p. 14).

ordinary Sanskrit fluency, grammatical expertise and formal method is one reason many Sanskritists are averse to characterizing Pāṇini's grammar as anything but *sui generis*.

Given that caveat, rule formulation and application themselves use concepts formulated next in modern times, particularly through the formalisms of modern mathematical logic. First is that of step-wise derivations, with those steps driven by rules operating on categories defined over initial discrete symbol sets, as noted. Then, most importantly, rules are codified using what we today think of as 'auxiliary markers' (or 'non-terminals'), simply additional metalinguistic signs whose role is to control the derivational process: what to do if a stem is marked as a past tense verb, what to do if a noun is marked as an instrumental object, how to indicate passive versus active, what sound adjustments to make for adjacent phonemes, and so forth. These auxiliary markers (*anubandha*), identified by Pāṇini using the term **it** (using boldface for markers and defined terms), are almost always appended to intermediate strings as affixes (i.e. *string^affix*) and retained as long as needed, or until the marker is changed or deleted in the derivation.[10] The term **it** derives from the Sanskrit particle *iti*, used as a quotation marker, and whose deictic status is reflected in allied terms such as *idam/this*, *iha/here*, *idānīm/now* (*Staal 1975*, p. 345). A derivation concludes with application of many phonological rules which convert an expression so that it is ready for speech, particularly through use of *sandhi* rules for adjusting adjacent sound forms, comparable to pronouncing the plural *boy* + *s* as */boyz/*. As mentioned above, Indian linguistics long recognized the discrete terms used in their analysis as abstractions; hence derived expressions required (internal and between-word) 'smoothing' to better approximate empirical speech. The last auxiliary markers for a set of derived words may be deleted, resulting in the finished Sanskrit sentence (with case endings and inflections basically dictate sentence structure), akin to a proved theorem or computer program. Alternatively, a set of final markers may be retained so that, among other uses, derived words may be recursively used as components in one of many types of Sanskrit compounds, a major focus of all Sanskrit linguistics, not only Pāṇini's. A derived *pada* with its last marking *retained* is available for assignment to a *kāraka* role to initiate a new derivation of either a new word or sentence. In the grammar, *pada* certifies a word derived via the rules, and specifically as ending with exactly one of two suffix types, s**up** or *ti***ṅ**, a 'suffix' being itself defined as a 'following element'. With *ti***ṅ,** there are two subsets of first-second-and third-person endings whose number is either singular, dual or plural (hence $2 \times 3 \times 3 = 18$ total). Similarly *s***up** represents a class of 21 nominal endings organized as seven cases × singular-dual-plural.

Here is a sketch of a sentence derivation (*Sharma 1987*, ch. 3; *Gillon 2007*). Suppose the goal is to derive a correct Sanskrit expression of 'Devadatta is cooking rice in a pot with firewood for Yajñadatta': *devadatta odanam yajñadattāya sthālyām kāṣṭhaiḥ pacati*. The *kāraka* roles chosen would be the verbal *action* of cooking, an *agent* Devadatta, a *patient* of the action which is rice, an *instrument* of firewood, a *location* of the pot, and a *recipient* Yajñadatta of the action. The *kāraka* categories are formally defined and regulated by

[10] Before Pāṇini, ritual theorists described sound changes and combinations in words, also using grammatical case endings as markers for combined phonemes, but limited to concrete examples, not defined categories using specialized nomenclature as occurred later. Categories of ritual acts or participants *were* described using special terms, and these streamlined the representation of ritual processes in oral recitation. Without such abbreviatory devices, long procedural descriptions would be laboriously repeated, thereby undermining the goal of compact and generic characterizations of ritual structure (*Staal 1990*, ch. 26). With Pāṇini, technical terms are extended to language as a whole, making possible an extension of the generative methods seen in ritual formulations. The use of case endings as markers is similarly adopted, but now to linguistic categories, and not just individual words. The sūtra style of abbreviated summary, and the paired notions of sūtra/rule and *paribhāṣā*/metarule, are also taken over by Pāṇini and his tradition to perfect the metalinguistic analysis begun by the ritual theorists, for whom exact linguistic expression was only one component of procedural correctness.

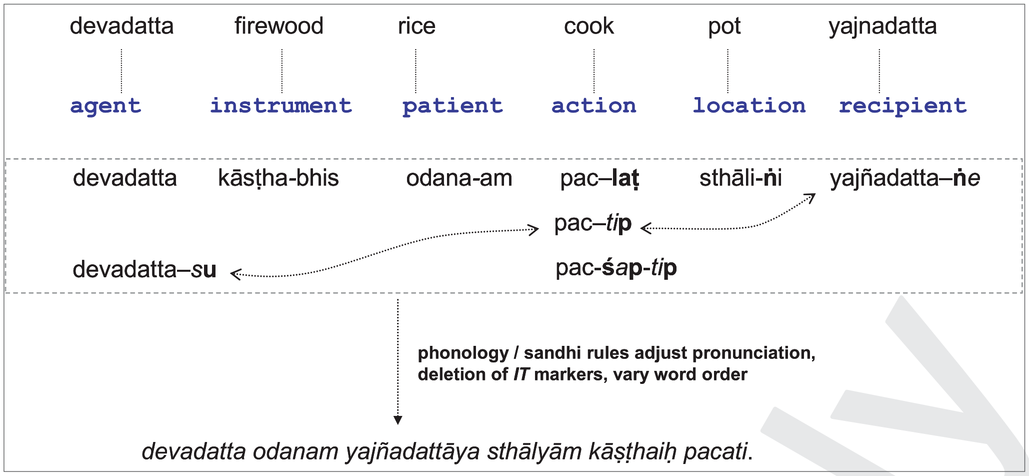

Figure 1. Schematic of derivation steps.

metarules, and provide a powerful heuristic for constructing a wide range of sentences. The categories mediate informal semantic meaning through their functional syntactic role. The free word order of Sanskrit means the selection initial stems, roots or words to associate with *kāraka* roles can be thought of as an unordered set: {*devadatta, firewood, rice, cooking, pot, yajñadatta*}, i.e. {*devadatta, kāsṭha, odana, pac, sthāli, yajñadatta*}. Again, as put Kiparsky, the grammar is a 'pure form of lexicalism' (*Kiparksy* 2009, p. 49).

These elements, schematized in Figure 1, now require rule applications to mark their assigned *kāraka* roles and to create new expressions. For example, the pot is singular and the location of the action, and that is marked by the suffix –**ṅ***i*, producing *sthāli*-**ṅ***i*. The boldface **ṅ** represents an auxiliary and non-terminal marker used in the derivation process, with italicized *i* being a terminal sound, and the hyphen indicating concatenation. Similarly, *yajñadatta* is the recipient of the action, marked by the suffix –**ṅ***e* and yielding *yajñadatta*–**ṅ***e*. The patient and instrumental roles, for rice and firewood respectively, can be marked with suffixes not needing auxiliary markers: *odana-am* and *kāsṭha-bhis*.

Derivation of the verb and its inflection for the cooking action, *pac*, involves more steps. There is first an assignment of the present tense using the suffix –**laṭ**, chosen from a set of **l** suffixes (*lakāra*) including perfect, imperfect, subjunctive, imperative, and other tenses. The verb can also refer actively to the agent Devadatta (*cooking the rice* ...), or passively to Devadatta by focusing on the rice (... *cooked by Devadatta*), an example of non-deterministic choice in the derivation. Devadatta is singular and is cooking for another, leading to the –**laṭ** suffix being replaced by –*ti***p**. The verb root *pac* also happens to require the vowel *a* between root and suffix, leading to *pac*-**ś***a***p**-*ti***p**. To now consistently mark the agent Devadatta as actively cooking, as planned with the active verb and required by the marker ti**p**, means use of the –*s***u** suffix on *devadatta*. The marked-up roles lead to {*devadatta*–*s***u**, *kāsṭha-bhis, odana-am, pac-* **ś***a***p**-*ti***p**, *sthāli*–**ṅ***i, yajñadatta*–**ṅ***e*}. An important caveat: each 'step' involves several substeps to identify the operational rule which can actually be applied, with a typical derivation citing all rules certifying a single rewrite step.[11] The substeps may involve numerous cross-references in the grammar

[11] 'In essence, operational rules cannot apply unless their interpretational or definitional rules are coupled with them. This can only be accomplished by the device of reference which is triggered by encountering a technical term or its denotation in an operational rule. This device reconstructs the term origin which, in turn, yields a referential index and it is this index that retrieves necessary information, explication or constraints relative to rule application' (*Sharma* 1987, p. 68).

or the application of metarules to resolve rule conflicts, possibly extending across several of the grammar's eight 'books'.[12] Pāṇini's complex derivations in this way differ significantly from those found in most modern formalisms. Nonetheless, once so obtained, rule application is largely a formal procedure.

Given Panini's process, intermediate expressions can be used to derive actual words by deletion of non-terminal markers, here **u**, **ś**, **p**, **ṅ**; that is finally followed by derivation of correct terminal sounds through phonological rules adjusting for transitions between phonemes, such as a plural *boy* + *s* pronounced */boyz/* but *cat* + *s* pronounced */cats/*.

The derivation above is typical in that auxiliary markers are similarly used throughout the grammar for rule expression and their application. Sanskrit syntax is already highly governed by case endings, so this basic means by which Pāṇini's grammar extends the Sanskrit object language by its own means. The systematic role for affixing makes Pāṇini's innovation a kind of *grammaticalization*, which is often central to language change generally.[13] Here, existing affixing resources of the object language, Sanskrit, are generalized to describe the object language itself. In modern computational theory, the analogous bootstrapping innovation is to use Post-type rewrite rules to formulate a metarule, or system, for all rewrite rules; or to use Turing machine grammar to define a universal Turing machine; or as shown by Gödel, to use number theory to define a metalanguage for its own derivations; and so forth. Such bootstrapping also occurs practically when a programing language like C + + or Pascal is used to write its own compiler, with successive versions accommodating larger swaths of the language. With Pāṇini, though, what differs is that we assume the spoken natural language Sanskrit and its structure to begin with. The object language is neither a mathematical invention nor is it necessarily even written.

Here is a related and final example of Pāṇini's care in distinguishing what we call in modern terms, following Frege, the *use* and *mention* of symbols or linguistic categories. Pāṇini throughout the grammar clearly distinguishes his object-level Sanskrit from the metalinguistic rules capable of generating it. As mentioned above, a Sanskrit particle *iti* is used to mark quotation, related to the technical term **it** for Pāṇini's auxiliary markers. Since the grammar abounds with object language stems, roots, and case endings, and these are mentioned rather than used, a morass of *iti* markers might seem inevitable. Instead Pāṇini reverses normal usage and makes *mention* of Sanskrit expressions his default, e.g. as represented in our writing by italics, *a*, *e*, *i*, *o*, *u*. The Sanskritist John Brough in 1951 identified

[12] For example, Book I includes global rules to which a derivation returns once other domains have been exhausted. Compounds are covered in Book II, while the recursive apparatus needed to form their inputs, two previously derived *padas*, is covered in subsequent books. Rules occurring later generally take precedence over earlier and typically more general rules. A complete rule may use multiple sūtras to address sub-conditions and exceptions. Header sūtras (*adhikārasutra*) announce a new section, identified through a special accent (*svarita*) which is not always recognizable. The sūtras following share assumptions which then need *not* be restated, because their 'recurrence' (*anuvṛtti*) is assumed from the header onward. That makes for complexity, but the gain is fewer sūtras codifying more conditions. Such organization follows ordinary references such as *Sally drove home, went in and turned on the television,* with the agent *Sally* remaining implicit. Sūtras can be marked to indicate their scope as applying to rules enumerated later, for which they should be assumed. Many rules, even when spelled out, may also require judgment for their proper application. That includes compound formation, in which input words are expected to be 'suitable' candidates relative to one another when combined. One might nonetheless form a strange compound, but not expect it to be heard in Sanskrit usage, analogous to Chomsky's *Colorless green ideas sleep furiously*.

[13] Through grammaticalization (*Brinton and Traugott 2005*, ch. 1) speakers may collectively and gradually modify content words, such as Old English *willan* (*to want*), as only a psychological state of desire, to become *will*, which today in English acts as a function word marking future intention (*Aitchison 2001*, ch. 7). More radically, the erosion and loss of Old English case endings and inflections led to the appearance of modern linear word order to identify roles earlier marked by affixes, which also occurred in the formation of French from Latin. With Pāṇini, affixing of the object language itself gets grammaticalized as a metalinguistic abstraction. On Sanskrit as object language and a basis for the metalanguage extending it, see *Houben 2009* (sec. 2) and *Staal 1995* (p. 107).

this assumption in sūtra 1.1.68 as stating that 'a word in a grammatical rule which is not a technical term denotes its own form' (*Brough* 1951, p. 403). Pāṇini's approach is again rediscovered millennia later. In 1963 the Algol 60 report describing the programing language, and known for use of Backus-Naur form in programing language design, explains that 'Any mark in a formula which is not a variable or a connective, *denotes itself* (or the class of marks which are similar to it)' (*Naur 1976*, p. 47, emphasis added).[14] Q3

3. Pāṇinian computation

The basic claim then is that Pāṇini's system is sufficiently structured to qualify as the earliest known computing language, in the sense defined at the start of this paper. Here are supporting details, along with several caveats. The idea of metalanguage and object language is understood by Pāṇini, and is present throughout his approach. The *paribhāṣā* metarules elaborate *how* the system is to be used, which is to apply operational rules to increasingly transformed symbolic expressions. As illustrated by the example, the principle method to express rules and rule application is through the auxiliary markers. That central technique means Pāṇini first devised the method of rewrite systems discovered by Emil Post starting in the 1920s but not recognized through publication until the 1930s (*Urquhart* 2009, sec. 4). The method is very much here, with even more formal rigor than found in Euclid's derivations.

Pāṇini goes so far as to devise a quite general formulation of a method he applies repeatedly, especially in his phonological rules, that of *context-sensitive* rules. We express those today as, e.g. $A \rightarrow B\ /C\_\_D$, meaning: replace expression A by B when A follows C and precedes D, with C or D possibly empty – so deletions can be treated as a kind of replacement (as done by Pāṇini). Pāṇini formulates an equivalent notion of generic string positions and their roles, perhaps his most elaborate formal construct which is directly comparable to a modern equivalent. So it is not possible to argue that Pāṇini has some serendipitous notion of rewrite rules; to the contrary, he developed the first expression of a central idea of linguistic rule types initiated by Chomsky and others in the 1950s, himself building on Post's ideas (*Pullum* 2011, sec. 2).

Here is an illustration of how Pāṇini's formal terminology works for a phonological rule, perhaps noted earliest as a context-sensitive formulation by *Staal* 1965b. The rule is to replace *i* by *y* when followed by any of nine Sanskrit vowels. That could be expressed as nine separate rules, but is better codified by a single rule referring to a right-context D of 'all following vowels' (and null left-context C). Similar replacements $u \rightarrow w$, *ṛ* → *r*, *ḷ* → *l* occur, again with any following vowel. In modern terms, this means a summary rule to be codified is the *ordered* replacement ⟨*i, u, ṛ, ḷ*⟩ → ⟨*y, v, r, l*⟩ when followed by a vowel, meaning a phoneme from the list {*a, i, u, r, l, e, o, ai, au*}. This list and others are coded as sublists in the *Śivasūtra* phoneme set, interpreted as being ordered as 14 separate 'rows'. Sublists of phonemes are identified by auxiliary markers for 'start' and 'endpoints', with those markers skipped or deleted in the sublist enumeration; that guidance is also spelled out as a metarule. It is also worth noting that Pāṇini's phoneme set, the *Śivasūtras* (suggesting deliverance by the god *Śiva*), while nominally expressed as a sequence of linear sūtras, is apparently optimally designed to enable its systematic reference to some 42 sublists of vowels and consonants.[15] The sounds and basic phonetics

[14] Algol 60 was the first programing language to be designed and described using formalized rewrite rules (*Preiestley* 2011, ch. 8). See also note 2 on Backus and Algol.

[15] The *Śivasūtras* imply 292 possible abbreviations (*prātyāharas*) of which Pāṇini utilizes 42. On the optimality properties of this organization, see *Petersen* 2004. The matrix-like complexity of the *Śivasūtras* alone is suggestive of inscriptional help.

Q15

Table 1.

	Articulation moving from back to front of mouth				
	Velar	*Palatal*	*Retroflex*	*Dental*	*Labial*
From unaspirated	*k*	*c*	*ṭ*	*t*	*p*
to aspirated	*kh*	*ch*	*ṭh*	*th*	*ph*
unvoiced and	*g*	*j*	*ḍ*	*d*	*b*
voiced consonants	*gh*	*jh*	*ḍh*	*dh*	*bh*
to nasals	*ṅ*	*Ñ*	*ṇ*	*n*	*m*

Notes: Diacritics represent sound differences, e.g. as created by tongue placement in the retroflex *ṇ*. After *Staal* 2008 (Figure 24).

associated with linguistic phonemes are assumed known by the grammar's user (*Cardona* 1988, p. 166) as illustrated by Table 1:

Table 1 uses place and manner of articulation in the vocal apparatus as two dimensions defining the discrete sound forms needed for grammatical analysis. This older 5 × 5 Vedic table is *not* the form used by Pāṇini, but already shows considerable knowledge of phonemic structure. Rows and columns are ancient predecessors of today's 'distinctive features', or dimensions characterizing a language's allowed phonemes. Relevant to computation, such is the grammar's 'hardware', or media realization, as speech rather than inscription. Here then is the organization of Sanskrit phonemes as the *Śivasūtras*:

> *a i u* **ṇ** / *ṛ ḷ* **k** / *e o* **ṅ** / *ai au* **c** / *ha ya va ra* **ṭ** / *la* **ṇ** / *ña ma ṅa ṇa na* **m** /
>
> *jha bha* **ñ** / *gha ḍha dha* **ṣ** / *ja ba ga ḍa da* **ś** /
>
> *kha pha cha ṭha tha ca ṭa ta* **v** / *ka pa* **y** / *śa ṣa sa* **r** / *ha* **l** //

This 14-row ordering can be used to select sublists by identifying start and end points, with auxiliary markers, in bold, being skipped or deleted in the sublist enumeration. So, in the *Śivasūtras*, *i***k** stands for {*i, u, ṛ, ḷ* } and a**c** refers to {*a, i, u, ṛ , ḷ , e, o, ai, au*}, which is the vowel list needed above; the braces { . . . } are our written convention. The other list needed is *ya***ṇ** or {*y, w, r, l*}. A sūtra applies for taking same-sized pairs of lists as ordered sequences instead of unordered sets; the rule basically allows the definition of finite mappings between defined lists.

Given names for desired lists (e.g. *i***k**, *a***c**, *ya***ṇ**), the second step is using them to construct a context-sensitive rule $A \rightarrow B \,/\, C \_\_ D$. The challenge then is to define these functional roles for *A*, *B*, *C*, *D*. Pāṇini's solution is to give the lists, through their names (*i***k**, *a***c**, *ya***ṇ**), *kāraka* case endings in a sūtra statement, and thereby to grammaticalize the rule. That is, the case endings are added to the names of the lists, treated as syntactic objects, to contextually define their roles in stating a context-sensitive rule.

Pāṇini's artificial case endings are therefore used to express 'in the place of *A*, substitute *B*, when after *C* and *D* follows', using several metarules: *genitive* case ending marks *A* as what is to be substituted; *nominative* case ending marks *B* to substitute for *A*; *ablative* case ending marks a preceding segment *C*; *locative* case ending marks a trailing segment *D*. The context-sensitive conventions also are used as a master format for sūtra coding and hence are a consistent clue to their meaning. That is, many operational rules are have their sūtra form framed as $A_{Genitive}\ B_{Nominative}\ C_{Ablative}\ D_{Locative}$. This paradigm simplifies the

See also *Staal* 1995 (p. 103) on an early notion of 'homorganic', or 'having the same place, producing organ and effort of articulation in the mouth': *samānasthānakaraṇāsyaprayatnaḥ*.

statement and recognition of grammatical rules in abbreviated sūtra form. Hence Pāṇini's 'operational rules are generally [context-sensitive] substitution rules' (*Joshi* 2009, p. 3).[16]

In the (well-known) example, the rule leads to: *i***k** + genitive, *ya***ṇ** + nominative, *a***c** + locative, or {*ikaḥ*, *yaṇ*, *aci*}. When the words are combined in that order, a sound-changing *sandhi* rule completes the derivation as *iko yaṇ aci*, sūtra 6.1.77. The rule in effect is a metalinguistic sentence which is meaningless in Sanskrit proper. That is a remarkable use of Sanskrit to bootstrap itself into a metalanguage. *Given* a rule like *iko yaṇ aci*, its case marking has to be decoded, which can be written as $[\textit{iko}]_{\mathrm{gen}}$ $[\textit{yaṇ}]_{\mathrm{nom}}$ $[\textit{aci}]_{\mathrm{loc}}$. From that, a skilled user may use the case endings to unwind the sūtra into the named lists and ordered mappings. The grammar does not perform that task for you.[17]

The clarity and directness of Pāṇini's system also comes at considerable cost. There are thousands of rules and metarules, organized as the grammar's eight books. The dependencies across rules, and the organization of rules into subgroups controlled by marked 'headings', are highly complex. The system should be thought of as *containing* a rigorous rewrite formalism, especially via the metalanguage through which rules are expressed, but with the grammar as a whole organized using many intricate linking, structural and referential devices. Critically, rules are codified as its thousands of brief and memorizable sūtras, which should *not* be identified with grammatical rules themselves. Pāṇini's overall motivation is one of economic efficiency (*lāghava*), rationally motivated by the oral codification of the grammar (*Kiparsky* 2005, p. 65; 2007).[18] Sūtras are versified mnemonics for a grammar which today has to be reconstructed from commentary dating from some few, to many hundreds, of years following Pāṇini. The grammar's rules are glossed in summary form, or *vṛtti*, and other commentary styles, spelling out tacit assumptions and the intended content of terse sūtra formulations.[19] That re-expression should not be taken to marginalize the sūtras' linguistic function. The sūtras demonstrate the most elaborate means by which large amounts of information – the grammar's generative rules, matrices of case endings, linguistic categories – can be compactly communicated given constraints of human memory, oral transmission and Sanskrit structure.

Hence Pāṇini's mnemonic sūtras have been decoded and recoded through the ongoing oral tradition of grammarians, who have evolved nomenclature and guidelines for stating Pāṇini's rules in explicit form, along with examples, variant interpretations and criticism. Whether Pāṇini's grammar was originally *formulated* without inscriptional aids is unknown, controversial, and very doubtful for some (*Goody* 1987, ch. 4). However, even assuming considerable inscriptional help, the finished product is highly refined and ready for oral expression by communities of experts. The grammar makes almost no explicit reference to inscribed signs, with the *Śivasūtra* phoneme set identified through the place and manner of sound formation in the vocal apparatus. Pāṇini's grammar itself, both in

16 Rule types include: introduced definitions (*saṃjñā*), such as types of compounds, tenses, cases and others; derivational or operational (*vidhi*) rules, whose application to user-selected inputs leads stepwise to Sanskrit sentences by rewriting successive expressions; metarules/*paribhāṣās* defining and guiding operational rules through rule precedence, resolution of rule clashes, default assignments, rule options, constraints and other guidelines for the grammar's use; and heading rules (*adhikāra*), organizing rule groups into topical domains subject to shared constraints.

17 The grammar is 'monotone', meaning not all derivations are reversible, hence no algorithm is implied for grammatical marking working backward from a phonemic representation (*Kiparsky* 2009, p. 36).

18 Efficiency, or 'economy', is a driving factor in much language change (*Aitchison* 2001, ch. 11; *Deutscher* 2005, p. 88), with the difference here, in the construction of a formal grammar, being that its formulations, as an extension of Sanskrit itself, are pursued methodically and consciously.

19 For example, sūtra 3.1.97 *aco yat* is glossed as 'affix *ya***t** occurs after a verbal root which ends in a vowel [*a***c**]'. Or 3.2.1 *karmany an*: 'affix *a***n** occurs after a verbal root which co-occurs with a *pada* denoting an object [*karman*]' (*Sharma* 1987, p. 81). The reference *a***c** is to a codified *list* of vowels as discussed *above*. On difficulties of interpreting grammatical affixing through oral transmission, see *Cardona* 1988 (pp. 54ff., 115).

complexity and organization, is like a user's manual for an early programing language including the language definitions themselves, all expressed for using speech as phonemic 'hardware'.

Let us return to our main point, regarding the implicit computing power of Pāṇini's system. In terms of the techniques Pāṇini needs and explicitly uses for his linguistic theory, the computational 'maximum' is that of context-sensitive rules, which are *not* sufficiently general to represent all computable functions (*Davis et al. 1994*, ch. 11). Hence the computational power of the grammar proper falls short of universal computation, while that limitation does not hold for the metalanguage through which the Sanskrit 'application' is defined. The flexibility of context-sensitive rules makes them easier to formulate and interpret, but for some, this rule type is already overkill for representing natural language syntax (*Pullum and Gazdar 1982*), which largely is Pāṇini's scope too. Leaving aside the issue of the 'right' level of algorithmic power needed for natural language grammars, it is nonetheless a simple observation, given the explicitness of Pāṇini's metalanguage, that his grammar can be directly *extended*, using his same method of auxiliary markers, to represent any rewrite system desired. For Emil Post's achievement was to show that it was just the *method* of auxiliary markers which could be used to simulate the derivations of any rewrite system. So that must be true of Pāṇini's system as well.

In general, a rewrite rule can be of the form $r\colon g_0X_1g_1 \ \ldots\ X_ng_n \rightarrow h_0Y_1h_1Y_2 \ \ldots\ Y_mh_m$. The g's and h's are fixed strings of symbols from a finite symbol set S, including null strings. Each X_i is an arbitrary variable string over S. The Y's can be any of the X's including repetitions. Post showed that productions generated by rewrite rules over a symbol set S could be reproduced in a canonical way by extending S to an $S*$ including new auxiliary symbols, and using metarules R_{S*} and standardized axioms A_{S*}, allowing new symbols from $S*$. This is much as Pāṇini's uses his case affixes, namely to control production of exactly the target language $L_{\boldsymbol{R}}$, with Post showing that his normal form is completely general. Post's canonical rules are all even in 'affixing' form $gX \rightarrow Xh$, with X a variable symbol and g and h fixed strings.[20] Post's methods show, for example, that axiom systems with rules such as $\{A,\ \textit{if } A \textit{ then } B\} \rightarrow B$, can also be reproduced in rewrite form, with Post himself noting applicability to Russell and Whitehead's *Principia Mathematica*. Post has no linguistic rules, just algorithmic ones, and Panini's grammar as a whole is poorly thought of as a Post system *simpliciter*. But the modern rewrite notation for expressions, $E \rightarrow E'$, is not anachronistic for Pāṇini's rule application, nor is the notion of a succession of rule applications by which a produced expression is derived.

Using his canonical form for rewrite systems, Post demonstrated that his production/rewrite systems are equivalent to the representational power of Turing machines by showing that his standardized rules can be used to enumerate all rewrite rules and their productions. So, modulo use of spoken phonemes rather than inscripted graphemes, according to Post's theorems Pāṇini's methods can also be used to represent any target language $L_{\boldsymbol{R}}$, and are in that way *capable* of expressing universal computation through a largely direct application of the grammar's rule system. The basic reason is that, as with a modern computing language, Pāṇini has a systematic method for introducing new symbolic categories and rules applying to those categories. It is not as if Pāṇini devised a single ad hoc rewrite system of ambiguous generality, something like a missionary grammar. Pāṇini completed the difficult metalinguistic work needed to lay the groundwork for universal

[20] The scare quotes are meant as a reminder that terms such as 'prefix' and 'suffix', like 'place' and 'position', are spatial metaphors, since an abstraction has no 'pre', 'post' or any other location. Pāṇini uses words for 'earlier' and 'later' in context-sensitive rules (*Scharfe 2009*, p. 29) with visual metaphors used elsewhere. For Post's normal form analysis, see *Post 1943*, *Minsky 1967* (ch. 13).

computation through a *Sanskrit** as his grammar, described through the grammar's metalanguage. Pāṇini is actually close to formulating a computing paradigm, given his generic and rigorous formulation of context-sensitive rules, but as mentioned, this rule class falls short of universal computation.

As another way of seeing the power of Pāṇini's methods, a thought experiment can be formulated to complete the steps by which a *Sanskrit*** is constructed on top of Pāṇini's *Sanskrit** which then affords universal computation. To construct such a *Sanskrit***, one might introduce a *kāraka* role which, by some new affixing marker, segregates numerical or computing terms from ordinary Sanskrit. Derivations starting from this new starting point would be isolated from the original grammar. Categorical definitions, for 'numbers', 'constants', 'variables', 'addition', 'multiplication' and other logical and computing categories need would be introduced using *saṃjñā* style rules, just as Pāṇini introduces his own technical terms. These may involve simple recursive definitions typical of computing syntax for categories such as 'formulas', 'sentences', 'proofs' and 'theorems'. Applied to such categories, just as with a modern formalism, rules for new derivations could be cast as replacement rules of many kinds, including the context-sensitive rules described earlier. The expression of computations using Pāṇini's grammar can be almost just a transcription of a modern formalism into the grammar, enabled by its native techniques for categorical and rule definitions.

The difference in computing scope is an important feature of the thought experiment in which Pāṇini's grammar is extended to represent universal computation. Natural language can involve intricate syntactical or semantical constructs, but these are not computationally complex. The complexity of languages is captured rather by language-specific formulations involving word order, affixing, inflection, anaphor, long-distance dependencies and much else. So Pāṇini's grammar, however intricate the phonological, morphological, and syntactic relations defined there, still falls short of implying general multiplication, exponentiation and other computable functions, until that functionality is specifically introduced. Multiplicative arithmetic (including $+$ and $\times$, not $+$ alone) is undecidable, so its conceptualization should be seen as a cognitively bold and creative step. That includes more complex recursions than are implicit in natural language syntax without additional grammaticalizations or equivalent changes. Many, if not most, modern languages can be *used* to define any computational or mathematical theory at all, but that does not mean that arbitrary mathematical or computational content is implicit in their rules of grammatical formation.

The heavy lifting to define the metalinguistic framework is completed by Pāṇini, while he has just *limited* his target application to be the grammatical expressions of spoken Sanskrit.[21] For that, he needs a complex linguistic theory, and a precise metalanguage for codifying the grammar of his Sanskrit object language. So the computing power needed by Pāṇini is not 'universal', but everything is in place for a computing environment realized in Sanskrit speech. The *paribhāṣā* metarules enable definitions of linguistic categories with which most *vidhi*/operational rules are concerned. That structure implies the grammar is formally open-ended for category and operational rule formation, but limited by the goal of characterizing existing Sanskrit. Hence it is prima facie possible to use the grammar's metarules to formulate rules for numeric or computational tasks, or symbolic

[21] Limiting computational power of grammars was noted as early as *Chomsky 1963*, p. 359. On linguistic completeness and mathematical modesty, see *Culicover 2004*, *Pullum and Scholz 2005* and *Pullum 2011*. Another characterization of a language plus its grammatical rules, with minimal computational assumptions, is that of a 'structured inventory of symbolic units' (*Langacker 1999*, p. 73; *Tomasello 2003*, p. 105) with rules themselves being yet more constructions. For that approach applied to Pāṇini, see *Houben 2009*.

manipulations of formal systems generally. The basic historical observation is that Pāṇini's grammar, while not a formalism, includes a modern formalism constructed from the natural language it takes as its idealized object. In this way, Pāṇini's grammar, including its metalinguistic apparatus, includes the earliest known computing language, created by generalizing grammatical devices of Sanskrit itself. The parallel between a modern computing language, or formalism, and Pāṇini's grammar can be thought of in terms his four major functional components: (i) the rewrite formalism by which Sanskrit expressions are ultimately constructed; (ii) the metalinguistic *paribhāṣā* rules guiding those operations; (iii) the finite inventory of phonemes, stems and roots to which rules initially apply; (iv) the versified sūtras codifying all the rules in reduced form. The sūtra formulation of the grammar can be compared to a computing language summarized in a terse programing manual intended for a certain class of machines and their programers. Here the machine is that of oral recitation, not inscription, and the programers are ancient linguists or grammarians.

It can be hazardous to project contemporary mathematical ideas into its distant history, but much evidence shows that such is not the case here in attributing understanding of advanced formal methods to ancient Indian linguists, heirs to a centuries-long algorithmic traditions of ritual description and analysis.[22] Pāṇini shows broad mastery of Post's rewrite technique, including the formulation of context-sensitive rules as a special case, and an understanding of many critical metalinguistic steps. Neither Pāṇini nor his contemporaries had an idea of universal computation, nor would anyone else for millennia. Pāṇini's grammar as a whole is *not* a computing language, nor a Post rewrite system. Pāṇini is unique and *mostly* a linguist, with his formalism a handmaiden to that role. But Pāṇini practiced modern rewrite methods, in their basics, with facility; then Post, millennia later, rediscovering rewrite techniques, conjectured the method is completely general and proved that is so. Post's methods today are also no marginal curiosity, but are ubiquitous in programing theory and design.

4. The phonemic hypothesis

Given our emphasis on Pāṇini's grammar as an extension of spoken Sanskrit, a conclusion is in order on the role of inscription in Pāṇini's grammar and, more generally, in formalisms such as computing languages and some modern grammars.

The status of Pāṇini's grammar *vis à vis* ancient literacy versus orality has long been the subject of some debate. Staal proposed decades ago that linguistic concepts preceding Pāṇini's work were the product of an orally dominant culture (*Staal* 1990, pp. 37, 371), one lacking expertise with writing, though there is no fundamental evidence for just how the grammar was designed and refined. The complexity of the codified lists of the *Śivasūtras*, or the matrix-like *s***up** and *ti***ṅ** word classifications mentioned above, should make one wary of any judgment based on tradition alone. Dissenting from Staal, the anthropologist Jack Goody argued that stratified lists, tables and related analytic concepts of moderate complexity require inscriptional technology (whether arithmetical, linguistic or a mix) for facile manipulation and consistent accuracy (*Goody* 1987, ch. 10). Pāṇini's grammar, once created, *is* formulated for recitation and the expression of spoken Sanskrit. But it is still possible that writing played a critical role in the design and formulation of the grammar, left behind for ideological and institutional reasons regarding how the grammar was to be reproduced across generations and continued as a cultural enterprise.[23] A first question to ask is whether there are further reasons to believe that Pāṇini's grammar could or could

[22] See notes 5, 10 and text *above*.

[23] Sanskrit is 'a language of the gods in the world of men' (*Pollack* 2006, ch. 1), with *Vāc* in the *Rig Veda* being language *and* a goddess manifested through proper language use, attainable through rule-oriented methods to protect against incorrect

not have been developed as the product of an oral tradition with no or virtually no inscriptional skills. A second question is what the answer could mean for modern computation, given the expression of formal methods, in Pāṇini, as an extension of Sanskrit speech. The proposal will be that Pāṇini's grammar must have relied on alphabetic writing for its elaborate segmented structure. The reasons cited, from modern ideas about writing systems, will suggest the further conclusion that all generative computing languages, and modern symbolic formalisms built on a finite discrete symbolic inventory, implicitly involve principles of alphabetic writing.

A start toward seeing why Pāṇini's grammar almost certainly used inscriptional aids, specifically alphabetic writing, in its formulation comes from the study of the nonliterate oral traditions and the introduction of writing systems. In the West, Homer's *Iliad* and *Odyssey* are famous examples of extensive works originally learned through repeated performances lacking a standardized written form. For centuries before the Greeks adopted the Phoenician alphabet for their own language, around the eighth century BC, Greece had been a nonliterate culture. For works like Homer's, recitation and recreation was entirely different from how that can occur when writing is available. The classicist Milman Parry identified the mnemonic role for Homer's metrical structure and the use of standardized mnemonic formulas: the 'wine-dark sea', 'swift-footed Achilles', 'long-dressed' Helen and so forth. While not referring to a standardized text, varied Homeric 'epithets' could be interpolated as needed in poetic recitation, either to fit existing verses or as lead-ins for the singer's improvization (*Foley 1988*, ch. 2). This is one means through which the epics could be transmitted across generations, with some faithfulness, in spite of there being no independent 'original' version to refer to, only individual performances. Without operational criteria for defining what counts as the same or different in poetic verses, the mnemonic stability provided by writing appears to have had no equivalent substitute.

More radically, in their joint field work with bards from the former Yugoslavia, Parry and his student Albert Lord found that notions such as poetic lines, beginnings and endings, and even separate words, were not easily grasped by these nonliterate singers, who were also expert at reciting long traditional verse. What members of a literate culture readily identify as structural features of spoken language did not appear to be so easily available to those lacking experience with inscriptional methods. Relevant to Pāṇini, Lord argued for the difficulty, if not the impossibility, of formulating various grammatical categories and distinctions in a purely oral tradition for which segmented language patterns are not apparent. Lord says of the Yugoslav 'guslars' he and Parry lived with that:

> When asked what a word is, he will reply that he does not know, or he will give a sound group which may vary in length from what we call a word to an entire line of poetry, or even an entire song. The word for 'word' means an 'utterance'. When the singer is pressured then to say what a line is, he, whose chief claim to fame's that he traffics in lines of poetry, will be entirely baffled by the question, or he will say that since he has been dictating and has seen his utterances being written down, he has discovered what a line is, although he did not know it as such before, because he had never gone to school. (*Lord 1960*, p. 25)

The suggestion, followed up by later anthropologists such as Goody, is that writing provides cognitive support critical not just to consistent memorization and reproduction in the oral register, but to the formulation of segmented grammatical categories themselves,

expression. The idealized role for speech meant that linguistic theory itself had to be unpolluted by extra-Sanskrit and extra-oral ingredients, especially writing.

comparable to the bards' challenges with linguistic categories of 'lines', 'verses' or 'words'. That is evidently relevant to Pāṇini's grammar with its explicit reliance on discrete phonemic versions of continuous speech, not to mention more elaborate grammatical categorizations.

The Greeks devised their alphabet by adopting the Phoenicians' writing system, developed for their Semitic language, to the Indo-European sounds and grammatical patterns found in Greek. Until the Phoenicians (or perhaps related Semitic speakers), writing systems relied on signs for syllables or concepts directly, through syllabic signs or logograms, often combined together. These varied writing systems all involve some implicit grammatical theory of speech represented: through logograms that people, ordinary objects, categories and processes are represented through speech; and through syllabograms that speech is composed of linearly ordered molecular sound segments.[24] Until the alphabet, the implicit analytical units did not make exclusive use of phonemes, approximated by consonants and vowels as productive elementary sounds. For comparison, Sumerian writing is likely the earliest writing system for a whole language, and was weighted relatively heavily toward logograms, like to represent the sun, or also god and sky, with some phonographic representation of syllables.[25] Sumerian was agglutinative, with most words monosyllabic, and suffixed or prefixed morphemes 'glued' directly onto a root word without additional inflectional changes. Hence this form of writing could work reasonably well for predominantly word-morpheme-syllabic sound groups, and even though that ended up requiring many hundreds of cuneiform signs. The writing was prominent for some 1300 years, up until about 1900 BC. Administrative or temple settings, and public proclamations, helped to infer logographic meanings inferred from context. When the Akkadians, who finally conquered Sumer, adapted Sumerian writing for their own inflected, and grammatically quite different, Semitic language, they were necessarily led to expand the inventory of syllabograms to represent many sounds impossible in Sumerian. For the Akkadians, the 'theory' of language did not fundamentally change and did not fit the new target language so very well either.

The Phoenicians' Semitic language used consonantal roots, similar to Arabic *ktb* for *write*, *srb/drink*, *qbr/ bury*. Roots are given syntactic roles by adding vowels 'inside', as in ***k**ā**t**i**b**/writer*, ***k**u**tt**ā**b**/writers*, ***k**a**t**a**b**a/he wrote*. Vowels are not important for signaling lexical differences, as English *dog/dig* or *ten/tan*, so only using consonants for writing could still work well in typical ancient communication contexts. Bi- or tri-consonants, like *br* and *spr*, could also be constructed without new signs, something not possible with a syllabary, and diacritical marks, like a superscript , were also used to mark vowels. In this way, the Phoenician system is close to a alphabet based on signs for consonants and vowels. With that different (implicit) linguistic model, the writing system can be adapted to sounds and sound groups of *any* spoken language, quite unlike a syllabary. The innovation of Phoenician consonantal/alphabetic writing was the implicit discovery that language sounds needed for *any* language can be represented as combinations of consonant and vowel sounds.[26] The problem of 'translation' experienced by the Akkadians was mostly dissolved. The alphabet, in this way, is the means by which duality of patterning, or the

24 On writing systems as implicit linguistic models, see *Coulmas 2003* (p. 139) and *Olson 1994* (ch. 12).

25 With the writing of whole sentences, some signs were also used as determinatives to mark some another sign's category: so signs for 'man' or 'woman' could determine the gender of a name, or using a sign for 'wood' added to a sign for 'plough' could indicate the tool rather ploughing as an activity. The rebus principle is a simple way to create phonograms, like using a picture of a sun to sound out */son/* in English.

26 *Gelb 1952* (ch. 4) argues that Phoenician writing *functioned* as a syllabary with vowels marked only in an 'irregular and sporadic fashion' (p. 182).

construction of meaningful language components from a structured finite set of meaningless elements, is first implicitly expressed as a universal and non-ad hoc linguistic principle.[27] The alphabet takes advantage of the 'recognized fact for millennia that there exists two complementary classes of speech sounds, consonants and vowels' (*Coulmas 2003*, p. 109). This is the linguistic reason alphabets are such a useful cognitive technology, and among the greatest inventions ever.

In contrast to neighboring Semitic languages, Greek needed vowels to discriminate many words, and ways to mark syntactic roles through both affixing and inflection. Vowels were also needed in starting positions of words, while that did not occur for Phoenician. The solution for the Greeks was to add signs for vowels, using some unneeded Phoenician letters where the associated Phoenician sounds did not appear in Greek. But having done that, what was true for the Greeks would hold for most, if not all, of the world's languages, meaning to represent phonemic structure using consonants and vowels as proxies for place and manner of sound formation, with the latter being understood by ancient Indian linguists as in Table 1. With that change, the Greeks' Phoenician letters (*phoinikeia grammata*), as they called their alphabet, is a writing system whose basis is formed by meaningless graphic signs mapped onto meaningless speech signs which approximate Greek phonology. Plato himself notes in the *Theaetetus* that the consonant *s* is '*a mere sound* like a hissing of the tongue. *B* again *has neither voice nor sound*, and that's true of most letters' (*Theaetetus* 203b in *Burnyeat 1990*, p. 340, emphasis added).

The C + V abstraction represents Greek or other speech sounds with sufficient accuracy for speech and its grammar to be efficiently bootstrapped into written form, then finessed as on its own terms using word breaks, punctuation and an amplified written grammar*, so to speak. Grammatical structure of the target language need no longer lead to hundreds of extra signs, as it did for Sumerians and Akkadians; nor some fundamentally new model of linguistic analysis on which to base a writing system. Alphabets work because natural languages, while 'conventional', are far from totally arbitrary in their sign sets. Speech sounds are limited by the human vocal apparatus, with Indian linguists apparently the first to explicitly identify their language's phonemes in that way. At the same time, the study of oral traditions suggests that the segmentation of speech patterns for an entire language needs a writing system for that segmentation to be reasonably successful. The Indian linguists, even before Pāṇini, also were conscious of the differences between continuous speech (*saṃhitā*) and its description through a grammar using discrete elements (*padapāṭha*). Hence the question of whether inscription was needed for constructing Pāṇini's grammar involves some of the most advanced linguistic knowledge found in the ancient world, far more than the 'lines' or 'words' of poetic verse.

For purposes of this paper, the proposal that writing systems are essential for the facile perception and manipulation of segmented phonemic patterns will be called the *phonemic hypothesis*. That narrows the proposals of Lord, Parry and Goody to a necessary role for alphabetic writing to conceptualize phonemes as linguistic building blocks. The power of alphabetic writing, as noted, is that it implicitly internalizes duality of patterning as realized across natural languages, namely through the approximation of a complete phoneme set using consonants and vowels. The phonemic hypothesis is a *converse* to that, meaning that, alphabetic inscriptional aids are needed to identify, classify and combine these sound forms with facility. The historical writing systems are examples of how writing involves some implicit model of language structure, through syllabograms, logograms, phonographic signs and their combined use. In all cases the representation of language is enabled by

[27] On duality of patterning and writing, see *Sampson 1985* (ch. 2).

fixing the essential temporal nature of speech through the spatial stability and comparative objectivity of graphic signs. But only an alphabetic rendering induces sufficient order through phonemic sound groups so that morphemes and other segmented grammatical patterns are readily transferred from speech to graphics. Alphabetic writing expresses duality of patterning by transferring the stability of inscribed signs to the transient and rapidly fading flow of speech, and then efficiently inheriting and amplifying spoken grammatical patterns back in script, whether in clay, papyrus or other media.[28] The phonemic hypothesis then is an alternative to direct evidence for the use of writing in the composition of Pāṇini's grammar, given the ubiquitous role of phonemes through the *Śivasūtras*, their roles as controlling and classificatory markers, and the detailed formulation of sandhi/phonological rules.

Three kinds of evidence support the phonemic hypothesis in addition to the historical and anthropological works mentioned. First is that discrete phonemic sounds have not been found to correlate with any physical signature in the waveforms of continuous speech. Isolated vowel or consonant sounds do show characteristic patterns, but in connected speech sounds are produced rapidly, influencing one another and causing boundaries to smear together. Upon hearing different instances of a word we can recognize them as 'the same', but the physical waveforms can be far from identical. For example, reversing a word's physical sound pattern, like that of *dog* to *god*, leads to an unintelligible result. Words differing only by a single sound, like /*cap*/ versus /*cab*/ will have physical differences spread further than at the single place changed (*Crystal 1987*, pp. 132ff.). Hence as recognized by Indian linguists, the characterization of speech into discrete segments is a considerable idealization of cognitively stereotyped sound patterns of continuous speech. The general finding is that it is 'impossible in general to disarticulate phonological representations into a string of non-overlapping units' (*Coulmas 2003*, p. 89).

A second type of evidence for the phonemic hypothesis includes controlled follow-up studies on the Lord-Parry field work. In a 1986 study, dozens of Portuguese adult volunteers, half of them nonliterate and the other half newly trained readers, were asked to play speech games involving removing or adding sounds to words. The results were that nonliterates failed in tasks requiring attention at the phoneme level, even though they could discriminate other speech sounds. Problems occurred, for example, when asked whether the same phonemes recurred at different places within words, or when asked to swap sounds in a word (*Morais et al. 1986*). Similar experiments show the difficulties nonliterates have in identifying morphemes and correct syntax (*Scholes and Willis 1991*). Third and finally as evidence for the phonemic hypothesis, there are clinical and neurophysiological theories relating phonemic awareness to reading ability. Stanislas Dehaene has argued that in child development, the mastery of letters and the understanding of phonemes

> are so tightly linked that it is impossible to tell which comes first, the grapheme or the phoneme – both arise together and enhance each other . . . .the relation between grapheme and phoneme development is probably one of constant reciprocal interaction . . . When we learn the alphabet, we acquire the new ability to carve speech into its elementary components. We become aware of the presence of phonemes in what initially sounded like a continuous speech stream. The well-read acquire a

[28] For duality, the *phone* [*p*] is a sound form regardless of its functional role in any language. The *phoneme* /*p*/ has an additional functional role in discriminating morphemes and words in the shared context of some language's phoneme set and its normative phonological rules. A given phone may be a phoneme in one language but not in another, or follow different rules if present in both.

universal phonemic code that facilitates the storage of speech sounds in memory, even if they are meaningless. (*Dehaene 2009*, pp. 202, 208)[29]

Dehaene's neuropsychological model is that facile reading of graphic signs relies on the brain's ability to 'recruit' neural areas associated with skills of line, edge and vertex pattern recognition which, outside of reading, facilitate physical object identification across a range of scales and spatial translations.[30] The evolution of reduced sign systems from pictograms, such as happened with cuneiform, exploits this ability through development of neural associations of graphic signs with spoken language elements and their grammatical or conceptual roles. This may explain reading deficits of dyslexia, reflected in single-word decoding, as due to impairments in grapheme–phoneme conversion. The positive conclusion, for Dehaene, is that it is a contingent fact of our plastic neural anatomy that we have writing systems, and facile recognition of phonemes, in any moderately sophisticated sense at all.

With this evidence for the phonemic hypothesis, the null hypothesis regarding Pāṇini's grammar should be that its formulation likely made use of alphabetic writing, even as the larger Indian culture was at the time nonliterate. Anything to the contrary would be a dramatic refutation of modern research on the psychology of phonemes and alphabetic writing. That is nonetheless consistent with the *finished* product, Pāṇini's grammar proper, being formulated for speakers facile with phonemic analysis of Sanskrit, and with speech being the grammar's home media and its sūtra-driven derivations.

As far as writing goes in India, the *Brāhm*ī script originated around 300 BC, so perhaps even the same century as Pāṇini's grammar. The script marks vowels using diacritics rather than letters, but its graphical design codes place and manner of sound articulation in the vocal apparatus, much as understood by Vedic linguists centuries earlier. Given that Greek alphabetic writing was invented by the eighth century, following by centuries well-developed writing systems and neo-alphabets of the Middle East, the use of Indian script several centuries later is not implausible. Even if Indian writing emerged independently from Semitic and Greek systems, early Indian linguists could have relied on it to model speech, later discarding the inscriptional apparatus when the oral formulation was sufficiently codified through versified and memorable sūtras. Presumably versions of the *vṛtti* commentary used today to interpret sūtras as rules existed in Pāṇini's time. The grammar's sūtras are 'first' in describing the grammar today but would have been 'last' in the grammar's construction as a compact distillation of rules in more useful *vṛtti* form.[31] Unfortunately we may never know just what inscriptional technology may have been used in ancient Indian linguistics, especially given the ideological rejection of writing as polluting. Even with that, it is possible that the use of graphic techniques was followed by a

[29] A main and subtle point of *Olson 1994* (p. 85) is to neither over- nor under-attribute the contribution of writing to phonemic awareness and understanding.

[30] These geometrical transformations are also ones for which we can recognize graphemes as representing the 'same' letter, such as through font size changes.

[31] *Deshpande 2011* addresses many internal features of Pāṇini's grammar relevant to possible inscriptional help *and* intrinsically oral formulations both, arguing that while some types of writing may have been known to Pāṇini and used as an aid, the grammar's oral conventions, particularly metalanguage markers using vowels or accents, would have been difficult to represent in script; see also *Scharfe 2009* (p. 69). For comparison, *Knox 1990* (pp. 16–19) argues that writing likely aided composition of our versions of the *Iliad*, even as the verses were still designed for recitation and using traditional metrical and mnemonic forms. The last oral versions were therefore augmented by inscriptional help, the word processing of its time. That conjecture, Knox argues, explains anomalies in epic structure and the implausibility of completely oral composition for the work we know. A comparable process may have occurred with the construction of Pāṇini's sūtras and their composition from early *vṛttis*. *Kiparsky 1980* (p. 240) suggests that Pāṇini's great interpreters, Kātyāyana and Patañjali, may have known the grammar 'not as living oral tradition, but in the form of a manuscript, and that whatever *vṛtti* went along with it was of secondary origin, and considered as such by them'.

purification ceremony accompanying memorialization in speech. All that notwithstanding, Pāṇini's grammar is *still* an organic extension of spoken Sanskrit, just one possible only *via* alphabetic writing for the conceptualization of phonemes, their new role as auxiliary markers, and the generative products so created.

In conclusion, we note the modern idea that computation can be expressed in any media you like, with software an abstraction independent of any hardware implementation. Pāṇini is *almost* an historical example of just that media freedom, as his grammar is formulated for orally expressed, spoken Sanskrit. But according to the phonemic hypothesis that oral formulation must have relied on lost inscriptional aids. A similar dependence of segmentation skills on the duality principles grounding alphabetic writing then must also be true of modern symbolic calculi, whether formal logics or computing languages. Modern computing languages, like structured grammars, require the tiered, hierarchical structures of symbolic forms found first in Pāṇini. That power requires a systematic approach to duality of patterning, like that of alphabets, which then can be applied to written language and formal systems too. The modern notion of a formal metalanguage requires the inherently metalinguistic tools of an alphabet or its equivalent to get started at all. This basis is taken for granted in Frege's *1879 Begriffsschrift*, or 'concept-script'[32]; in the classic computing Q4 paradigms of Post and Turing with their explicit inscriptional metaphors; and in computing languages and modern formal systems generally. Such a basis was almost surely used by Pāṇini, his grammar's formalism being the earliest historical example of the kind ubiquitous today in computer science and mathematical logic. Nonetheless, Pāṇini showed, by constructing a whole formal language through the affixing resources of the Sanskrit object language itself, that the differences between natural and artificial computing languages are smaller than often thought. Not because natural languages are, or are close to being, computing languages, but because the development of computing languages, whether ancient or modern, is a continuation of natural language constructions by their own means.

References

Aitchison, J. 2001. *Language Change: Progress or Decay?* 3rd ed., Cambridge: Cambridge University Press.

Backus, J. 1959. 'The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM conference', in *Proceedings of the International Conference on Information Processing*, Paris: UNESCO, June 1959, pp. 125–32.

Backus, J. 1980. 'Programming in the 1950s – some personal impressions', in N. Metropoliset al., *A History of Computing in the Twentieth Century*, New York: Academic Press.

Bloomfield, L. 1935. *Language*, London: George Allen & Unwin. Q5

Brinton, L., and Traugott, E. 2005. *Lexicalization and Language Change*, New York: Cambridge University Press.

Brough, J. 1951. 'Theories of general linguistics in the Sanskrit grammarians', in J. F. Staal, *A Reader in the Sanskrit Grammarians*, Cambridge, MA: MIT Press, pp. 402–14.

Burnyeat, M., ed. 1990. *The Theaetetus of Plato*, trans. M. J. Jevett, Cambridge: Hackett.

Cardona, G. 1988. *Pāṇini: His Work and Its Traditions*, **Vol. 1**, Delhi: Motilal Banarsidrass.

Cardona, G. 2009. 'On the structure of Pāṇini's system', in G. Huet, A. Kulkarni, and P. Scharf, *Sanskrit Computational Linguistics*, Lecture Notes 5402, Berlin: Springer-Verlag, pp. 1–32.

Chomsky, N. 1963. 'Formal properties of grammars', in R. D. Luceet al., *Handbook of Mathematical Psychology II*, New York: John Wiley and Sons.

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

Coulmas, F. 2003. *Writing Systems: An Introduction to Their Linguistic Analysis*, New York: Cambridge University Press.

Crystal, D. 1987. *The Cambridge Encyclopedia of Language*, New York: Cambridge University Press.

[32] *Frege 1997, 1879* (pp. 49–51) comments on the relation of his new *Begriffsschrift* to language, but not writing per se; the translation 'ideography' in *van Heijenoort 1967* (pp. 6–7) for *Begriffsschrift* is then potentially misleading. Frege's labeling his system a 'script' is wholly appropriate but can today be reconsidered in light of the historical development of writing systems and their linguistic analysis.

Culicover, P. 2004. 'The Cambridge English grammar', Review of R. Huddleston and G. Pullum, eds., *Language*, **80**(1), 127–41.

Davis, M., Sigal, R., and Weyuker, E. 1994. *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, 2nd ed., Boston, MA: Academic Press.

Dehaene, S. 2009. *Reading in the Brain: The Science and Evolution of a Human Invention*, New York: Viking.

Deshpande, M. M. 2011. 'From orality to writing: transmission and interpretation of Pāṇini's *Aṣṭādhyāyī*', in *Le Symposium International Le Livre. La Roumanie. L'Europe* 2010 vol. III, Bucharest: Éditeur Bibliothèque de Bucarest, pp. 57–100.

Deutscher, G. 2005. *The Unfolding of Language*, New York: Metropolitan Books.

Foley, J. M. 1988. *The Theory of Oral Composition: History and Methodology*, Bloomington: Indiana University Press.

Frege, G. 1997. 'Begriffsschrift', trans. M. Beaney, in M. Beaney, *The Frege Reader*, New York: Blackwell, pp. 47–78 (first published 1879 in German).

Gelb, I. J. 1952. *A Study of Writing*, rev. 1962, Chicago, IL: University of Chicago Press.

Gillon, B. 2007. 'Pāṇini's Aṣṭādhyāyī and linguistic theory', *Journal of Indian Philosophy*, **35**(5), 445–68.

Goody, J. 1987. *The Interface Between the Written and the Oral*, New York: Cambridge University Press.

van Heijenoort, J., ed. 1967. *From Frege to Gödel: A Sourcebook in Mathematical Logic*, Cambridge: Harvard University Press.

Horowitz, E., ed. 1983. *Programming Languages: A Grand Tour*, Rockville, MD: Computer Science Press.

Houben, J. E. M. 2009. 'Pāṇini's grammar and its computerization: a construction grammar approach', in A. Kulkarni and G. Huet, *Sanskrit Computational Linguistics,* Lecture Notes 5406, Berlin: Springer-Verlag, pp. 6–25.

Huet, G., Kulkarni, A., and Scharf, P., eds. 2009. *Sanskrit Computational Linguistics*, Lecture Notes 5402, Berlin: Springer-Verlag.

von Humboldt, W. 1999. *On Language: The Diversity of Human Language Construction and Its Influence on the Mental Development of the Human Species*, ed. M. Losonsky and trans. P. Heath, New York: Cambridge University Press (first published 1836 in German).

Ingerman, P. 1967. 'Pāṇini Backus form suggested', *Communications of the ACM*, **10**(3), 137.

Jakobson, R. 1990. *On Language*, eds. L. Waugh and M. Monville-Burston, Cambridge: Harvard University Press.

Joshi, S. D. 2009. 'Background of the *Aṣṭādhyāyī*', in G. Huet, A. Kulkarni, and P. Scharf, *Sanskrit Computational Linguistics*, Lecture Notes 5402, Berlin: Springer-Verlag, pp. 1–5.

Kadvany, J. 2007. 'Positional value and linguistic recursion', *Journal of Indian Philosophy*, **35**(5), 487–520.

Kiparsky, P. 1980. *Pāṇini as a Variationist*, ed. S. D. Joshi, Cambridge: MIT Press.

Kiparsky, P. 2005. 'Pāṇinian linguistics', in E. F. K. Koerner and R. E. Asher, *Concise History of the Language Sciences*, Cambridge: Pergamon/Cambridge University Press.

Kiparsky, P. 2007. 'Pāṇini's razor', Paris, Lecture Notes. http://web.stanford.edu/ kiparsky/

Kiparsky, P. 2009. 'On the architecture of Panini's grammar', in G. Huet, A. Kulkarni, and P. Scharf, *Sanskrit Computational Linguistics*, Lecture Notes 5402, Berlin: Springer-Verlag, pp. 33–94.

Knox, B. 1990. 'Introduction', in *The Iliad*, trans. R. Fagles, New York: Viking Penguin, pp. 3–64.

Kulkarni, A., and Huet, G., eds. 2009. *Sanskrit Computational Linguistics*, Lecture Notes 5406, Berlin: Springer-Verlag.

Langacker, R. W. 1999. *The Foundations of Cognitive Grammar: Volume I: Theoretical Prerequisites*, Stanford: Stanford University Press.

Lord, A. 1960. *The Singer of Tales*, Cambridge: Harvard University Press.

Minsky, M. 1967. *Computation: Finite and Infinite Machines*, New York: Prentice-Hall.

Mishra, A. 2009. 'Simulating the Pāṇinian system of Sanskrit grammar', in G. Huet, A. Kulkarni, and P. Scharf, *Sanskrit Computational Linguistics*, Lecture Notes 5402, Berlin: Springer-Verlag, pp. 127–38.

Morais, J., Bertelson, P., Cary, L., and Alegria, J. 1986. 'Literacy training and speech segmentation', *Cognition*, **24**(1–2), 45–64.

Naur, P.et al. 1963. 'Report on the algorithmic language ALGOL 60', in E. Horowitz, *Programming Languages: A Grand Tour*, Rockville, MD: Computer Science Press, pp. 44–60.

Olson, D. 1994. *The World on Paper*, New York: Cambridge University Press.

Olson, D., and Torrance, N., eds. 1991. *Literacy and Orality*, New York: Cambridge University Press.

Ostler, N. 2005. *Empires of the Word*, New York: Harper Collins.

Petersen, W. 2004. 'A mathematical analysis of Pāṇini's Śivasūtras', *Journal of Logic, Language and Information*, **13**(4), 471–89.

Pollack, S. 2006. *The Language of the Gods in the World of Men: Sanskrit, Culture, and Power in Premodern India*, Berkeley: University of California Press.

Post, E. 1943. 'Formal reductions of the general combinatorial decision problem', *American Journal of Mathematics*, **65**(2), 197–215.

Priestley, M. 2011. *A Science of Operations: Machines, Logic and the Invention of Programming*, London: Springer-Verlag.

Pullum, G. 2011. 'On the mathematical foundations of *Syntactic Structures*', *Journal of Logic, Language and Information*, **20**(3), 277–96.

Pullum, G., and Gazdar, G. 1982. ‘Natural languages and context-free languages’, *Linguistics and Philosophy*, **4**(4), 471–504.

Pullum, G., and Scholz, B. 2005. ‘Contrasting applications of logic in natural language syntactic description’, in Petr Hájek et al., *Logic, Methodology and Philosophy of Science: Proceedings of the Twelfth International Congress*, London: King’s College Publications, pp. 481–503.

Renou, L. 1941. ‘Les connexions entre le rituel et la grammaire en sanskrit’, in J. F. Staal, *A Reader in the Sanskrit Grammarians*, Cambridge, MA: MIT Press, pp. 435–69.

Sampson, G. 1985. *Writing Systems*, Stanford: Stanford University Press.

Scharf, P. 2009. ‘Modeling Pāṇinian grammar’, in G. Huet, A. Kulkarni, and P. Scharf, *Sanskrit Computational Linguistics*, Lecture Notes 5402, Berlin: Springer-Verlag, pp. 95–126.

Scharfe, H. 2009. ‘A new perspective on Pāṇini’, *Indologica Taurinensia: Online Journal of the International Association of Sanskrit Studies*, **35**, 1–270. http://www.indologica.com/index-volume35.asp

Scholes, R., and Willis, B. 1991. ‘Linguists, literacy and the intensionality of Marshall McLuhan’s Western man’, in D. Olson and N. Torrance, *Literacy and Orality*, New York: Cambridge University Press, pp. 215–35.

Sharma, R. N. 1987. *The Aṣṭādhyāyī of Pāṇini I: Introduction to the Aṣṭādhyāyī as a Grammatical Device*, New Delhi: Munshiram Manoharlal.

Staal, J. F. 1965a. ‘Euclid and Pāṇini’, in J. F. Staal, *Universals: Studies in Indian Logic and Linguistics*, Chicago, IL: University of Chicago Press, pp. 143–60.

Staal, J. F. 1965b. ‘Context-sensitive rules in Pāṇini’, in J. F. Staal, *Universals: Studies in Indian Logic and Linguistics*, Chicago, IL: University of Chicago Press, pp. 171–80.

Staal, J. F., ed., 1972. *A Reader in the Sanskrit Grammarians*, Cambridge, MA: MIT Press. Q14

Staal, J. F. 1975. ‘The concept of metalanguage and its Indian background’, *Journal of Indian Philosophy*, **3**(3), 315–54.

Staal, J. F. 1988. *Universals: Studies in Indian Logic and Linguistics*, Chicago, IL: 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**(1), 73–127.

Staal, J. F. 2008. *Discovering the Vedas: Origins, Mantras, Rituals, Insights*, Kundli: Penguin.

Tomasello, M. 2003. *Constructing a Language: A Usage-based Theory of Language Acquisition*, Cambridge, MA: Harvard University Press.

Urquhart, A. 2009. ‘Emil post’, in D. M. Gabbay and J. Woods, *Handbook of the History of Logic, Vol. 5: Logic from Russell to Church*, Amsterdam: North-Holland.

Williams, M. 1846/2005. *An Elementary Grammar of the Sanscrit Language*, New Delhi: Cosmo.