# Sanskrit signs and Pāṇinian scripts

**Gérard Huet**
Inria Paris Center,
France

## Abstract

We discuss ways of understanding the Pāṇinian grammatical tradition of Sanskrit in computationally tractable ways. We propose to dissociate the formal expression of the locutor's communicative intention (expressed as a composition of sign combinators called a *script*), from its justification (using Pāṇinian rules and meta-rules). Computation consists then in evaluating a Pāṇinian script to its final sign, delivering both the correct enunciation, and its meaning expressed as a non-ambiguous paraphrase.

## 1 Computational linguistics and the *Aṣṭādhyāyī*

It is now recognized as an undisputed fact that Pāṇini was a genius linguist 25 centuries before linguistics was established as a scientific discipline in Europe by de Saussure, and that his *Aṣṭādhyāyī* is a very complete and precise grammar of Sanskrit. This scholarly consensus must be distinguished from opinions stated in various social media, claiming that Pāṇini's *Aṣṭādhyāyī* is a faultless computer program, and that Sanskrit is the perfect programming language of the future. Usually such hyperbolic assertions (*atiśayokti*) are not backed up by any argumentative justification.

It has also been claimed that Pāṇini invented the Backus-Naur form of context-free grammars. This originates from a 1967 note in a computer journal by Peter Ingerman (Ingerman, 1967) without any precise evidence. Such uninformed anachronistic judgements are misleading, and just add confusion to the debate around the actual contribution of Pāṇini to formal computation and information theory besides linguistic modeling.

Actually, even if it is far-fetched to recognize a context-free grammar description in Pāṇini's grammar, it is a fact that many formal description mechanisms are explicit in the *Aṣṭādhyāyī*. For instance, external sandhi operations are defined by *sūtra*s of a standardized form which may be unambiguously decoded as algebraic rewrite rules of the form : $[x]u|v \to w$, with $x, u, v, w \in \Sigma^*$, where $\Sigma$ denotes the set of phonemes (*varṇa*) of Sanskrit. The encoding uses Sanskrit morphology (*vibhakti*) to discriminate the fields of a record encoding the 4-tuple of strings $x, u, v$ and $w$ that are the parameters to the rewrite rule (Cardona, 1974; Bhate and Kak, 1993). The rule may be read as a computation procedure to rewrite a juxtaposition of $u$ and $v$ in the input string as string $w$ in a left context $x$. That is, $XxuvY$ may be rewritten as $XxwY$ for any strings $X$ and $Y$. If we further specify that rewriting is done uniformly in a left-to-right fashion, we get indeed a *vikāra* algorithm (*vidhikalpa*) that applies (external) sandhi to strings of phonemes in order to transform a list of isolated words (*padapāṭha*) into a continuous enunciation (*saṃhitāpāṭha*). It is easy to relate such rules to contemporary morpho-phonetic rules in computational linguistics, building on the theory of regular relations in formal language theory (Kaplan and Kay, 1994; Koskenniemi, 1984). Indeed, such Pāṇinian rules may be directly fed into the finite state toolkits implementing this paradigm (Huet, 2005; Hyman, 2009). This sort of mechanism may be applied as well to vowel grade shift (*guṇa*, *vṛddhi*), vowel harmony, etc.

The situation is more complex for generative morphology, where word construction from morphemes and affixes uses retroflexion, which needs for its specification a non-regular operation, where the left context must be inspected on an unbounded, although generally small, suffix. Indeed, many Pāṇinian rules are of a more complex nature, involving context-free and even context-sensitive formulations. Furthermore, the "flow of control" of Pāṇinian rules, including rules of a meta-linguistic nature, is a complex affair, and it is not possible to regard *Aṣṭādhyāyī* directly as a computer program whose instructions would be the *sūtra*s. Actually, part of the problem is the conciseness (*lāghava*) of its description, a very important concern since the grammar had to be exactly memorized by the traditional students. We may rather think of *Aṣṭādhyāyī* as a high-level program *compiled* into a low-level machine code, where techniques of compaction such as sharing have been applied to obtain a low memory imprint, at the expense of control complexity. Indeed, the advent of printing allowed equivalent reformulations of the grammar in more hierarchical ways, and presumably of easier use to the student, but at the expense of duplication of rules (Dīkṣita et al., 1905).

It remains that Pāṇini is the ultimate authority, and that the perfection of its description induced a prescriptive nature of the grammar, seen as the gold standard of Sanskrit, following Patañjali magisterial commentary (Joshi and Roodbergen, 1990; Filliozat, 1975). This explains the stability of the language, since it could evolve only through the constraints of the grammar. Thus further commentaries were reduced to settle matters of details, and to elucidate the flow of control of the grammar usage (Sharma et al., 2008; Joshi and Roodbergen, 2004; Sharma, 1987).

Thus Pāṇini's *Aṣṭādhyāyī* is often (justly) referred as a *generative* grammar for Sanskrit. Actually, when challenged, a competent (*śiṣṭa*) Sanskrit locutor should be able to exhibit the sequence of Pāṇinian *sūtra*s (*prakriyā*) validating his linguistic productions. Indeed, such systematic sequences have been worked out for the various examples discussed in traditional grammars (Grimal et al., 2006). Thus it would seem that it could be possible in principle to write a simulator of Pāṇinian derivations which would take *sūtra*s as instructions and derive Sanskrit strings guaranteed by construction to be correct Sanskrit.

## 2 Using the *Aṣṭādhyāyī* in generation

There have been indeed attempts to write a simulator as a computer program that would progressively elaborate a target Sanskrit utterance as a sequence of operations on a string of phonemes – certain ending up as phonetic material, others being meta-linguistic markers (*anubandha*) which are progressively eliminated when the operation they trigger is effected. See for instance the work of Anand Mishra (Mishra, 2009; Mishra, 2010), of Peter Scharf (Scharf, 2009), and of Pawan Goyal et al. (Goyal et al., 2009).

The first remark to be made is that the *Aṣṭādhyāyī* is not self-sufficient. It must be used together with specialized lexicons, one giving roots with derivational markers (*dhātupāṭha*), another one giving lists of words sharing morphological characteristics (*gaṇapāṭha*), still other ones listing attested genders of substantives (*liṅgānuśāsana*) (Cardona, 1976). Access to these resources is triggered by root or stem selection. One practical problem is to decide which version of these resources to use, since the lexical lists are open-ended and have been amended or reorganised since Pāṇini's time.

Another difficulty is that checks must be effected that a rule application is indeed permitted at the time of its invocation. This induces the maintenance of complex data structures storing the derivation history, the verification of context conditions implicitly carried over from one *sūtra* to the next (*anuvṛtti*), but also the analysis of complex priority rules between *sūtra*s (*siddha*, *asiddhavat*) which are not always consensual among experts. Also, certain *sūtra*s are subject to semantic conditions (rule X is valid for root R "in the sense of ...") which are not directly amenable to computation. Aspects of this control problem, and their relation with computational devices, have been discussed in (Goyal et al., 2009).

Finally, many rules specifying optional operations are non-deterministic in nature (with a long history of discussions on the optionally/preferably interpretations (Kiparsky, 1980)).

These difficulties lead one to believe that *Aṣṭādhyāyī* can be used to generate an enunciation $S$ only if, not only $S$ is known in advance, but its intended meaning is known too. And there might still be choices in the application of rules which must be made explicit if one wants to obtain a deterministic simulation.

The rules discuss both forms and meanings. However the grammar cannot be construed to generate meaning from correct enunciations (think of *śleṣa* ambiguity), nor correct enunciations from meaning (since there are many ways to say the same thing, specially in a language with flexible word order). Rules have conditions both on the surface realisation (phonemic strings) of the considered enunciation and on its intended meaning. Any attempt to explain generativity in unidirectional way runs into circularities (*itaretarāśrayadoṣa*). As Peter Scharf puts it: "The rules do not actually generate the speech forms in certain meanings; they instruct one that it is correct to use certain speech forms in certain meanings" (Scharf, 2009).

The solution to these difficulties is to make explicit oracle decisions fixing all these choices[1], and to consider that the derivation process operates not just on surface material (strings of phonemes and markers) but on *signs* in the sense of de Saussure, that is pairs of enunciations and of their meanings. This will be possible if we identify precisely the semantic combinators implicit in the derivational process. The derivational process ought to derive not just the target final enunciation, but also a formal expression representing its sense, or some disjunction of possible senses, when some ambiguity remains.

## 3 Sanskrit signs

Ferdinand de Saussure, the Swiss scholar who created the Western linguistics discipline in the 19th century (with knowledge from Pāṇini and the *Vyākaraṇa* tradition) puts at a central place the notion of *sign*, as a pair [signifiant/signifié] associated with linguistic entities (de Saussure, 1916). The "signifiant" (meaningful enunciation: *śabda*) is paired with the "signifié" (its meaning: *artha*), and the relation between the two (*śabdārthasambandha*) is postulated to be an arbitrary artefact of each human language.

Scholars have argued that Bhartṛhari anticipated the Saussurian sign by considering language [*śabda*] under the two forms of articulated sound signal [*dhvani*] and meaning [*artha*] (Raja, 1969; Houben, 2002). But such discussions tend to involve psycho-linguistic notions such as the cognitive act of linguistic understanding [*sphoṭa*] which are of no relevance here. Still, the relationship between enunciation and meaning was articulated as the fundamental notion underlying meaningful speech since antiquity. Indeed, the first explanation (*vārttika*) by Kātyāyana quoted by Patañjali in his commentary Mahābhāṣya on the *Aṣṭādhyāyī* reads: *siddhe śabdārthasaṃbandhe* "The relation between the utterance and its meaning having been established". This has been amply commented, but it should be stressed that here the meaning component *artha* should be glossed as sense and not as denotation. The arbitraryness of the sign relation is here expressed by the "eternity" of language: this relation was known of all eternity (*nitya*), i.e. it is not a "created" artifact.

In order to make computational use of signs, we must be able to represent both sides of the sign as combinatorial objects. For the signifiant, the natural solution is to discretize the phonetic signals using the notion of phoneme. Actually, the work has been prepared by Pāṇini (or even perhaps his predecessors) since the *varṇamālā* is the Sanskrit alphabet of phonemes, conveniently ordered by the position and manner of articulation. Ancient grammarians of Sanskrit had perfectly understood the notion of phoneme, 25 centuries before Nikolai Trubetzkoi, usually credited in the West with the invention of the notion.

For the signifié, things are not so simple, since meaning is a complex notion, not readily axiomatizable in non-naive mathematical terms. We propose to use for its modeling a notion of non-ambiguous paraphrase, that is to replace semantics by non-ambiguous syntax. This is fully in the spirit of the *vyākaraṇa* tradition, where the meaning of enunciations is explained in terms of precise paraphrases. Actually the *Nyāya* discipline has pushed this technique to a very elaborate extent with a concept calculus called *Navyanyāya*.

In taking paraphrasing as the model for meaning, we are avoiding dealing with complex issues of denotation by 'objects' whose ontological nature is complex, and with quantification issues (the so-called problem of universals).

## 4 The semantic side of morphology

Verbs represent actions (including judgements). Substantives, qualified by adjectives, may represent actors (*kāraka*) of situations, or the situations themselves, as completed actions (*bhāva*). Actors fill roles of verbs, according to the valency table of their complements (*ākāṅkṣā*). This identification operates at

[1] The notion of oracle stems from non-deterministic computing; here the oracle decisions are the choices of the various linguistic constructions and of their parameters independently from their checking by the grammar.

two levels: words (morphology) and sentences (syntax). By syntax we mean more than constituency structure, but rather structural syntax in the sense of Tesnière (Tesnière, 1959), i.e. dependency graphs.

Let us first look at nominal morphology. The first level is *kṛt* formation, i.e. primary derivatives from verbs. For instance, participles, which have a definite uniform semantic character, that can be made explicit with a systematic paraphrase. But this also applies to most *kṛt* suffixes. A primary derivative mark (*kṛt*) is not just a notation for a suffix formed with phonetic material interspersed with markers (*anubandha*), so that the corresponding stem can be derived from the generating root. It also denotes a definite role (*sādhana*) with respect to the internal action associated with the root it derives from. This *sādhana* is often a *kāraka* role, such as Agent (*kartā*), but it also may stand for the Action itself (*bhāva*) (or more exactly to the state of affairs at the conclusion of the action). When the root is transitive, an agentive role will usually be conditioned on the concomitant production of the corresponding patient/goal (*karma*), typically as the left component of a compound word whose right component is the *kṛdanta*.

One can read the Pāṇinian *sūtra*s of section 3.1.133 to 3.3.17 as telling the story for agentive meaning, with 3.2.1 starting the treatment of transitive verbs. Similarly, the section 3.3.18 to 3.3.130 explains the (completed) action meaning or the other *kāraka*s. Here we have two difficulties. The first one is that the *sūtra*s are jumbled together in complex ways in order to give proper priorities, exceptions, blockings - the control structure. Thus a given *kṛt* suffix is not fully explained in a specific continuous section of the grammar, but may appear and reappear at various points. The second difficulty is that sometimes a precise *sādhana* is specified for a given suffix, but often it is under-specified (such as a non-agentive role). We shall solve this issue by demanding that a fully explicit *sādhana* be expressed when we produce the corresponding *kṛdanta* in the derivation, in order to construct a non-ambiguous semantic component, in the form of a specific paraphrase. Thus primary formation is hierarchized: Given a root and its valency, we choose a *sādhana*, then we choose an appropriate *kṛt* suffix, then we choose a *sūtra* justifying use of this *kṛt* suffix, then possibly we turn to the consideration of a co-occurring *upapada* in order to fulfill the remaining valency of the verb (*ākāṅkṣā*). Sometimes the construction is predicated of verbs endowed with specific preverbs (*upasarga*). Sometimes semantic constraints are specified (mode, frequency, tense, etc.)

This explains the stem formation. Then we complete the word formation by specifying its inflexion parameters, i.e. gender, number and case. This also is seen as an operation on the sign of the stem, since it affects both its form and its meaning.

Secondary word formation operates similarly, by similar interpretation of the *taddhita* suffix. Here, however, the great variety of such suffixes and their subtle semantic interpretations in the large *taddhita* section of the *Aṣṭādhyāyī* (covering 1115 rules starting with 4.1.76) merit a serious study in order to understand the proper granularity of the semantic combinators involved. In order to keep a manageably small set of distinct combinators, many suffixes will have to be considered synonyms. The monograph by Saroja Bhate (Bhate, 1989) can be the starting basis for this meticulous investigation. Furthermore, recent work in an object-oriented model of taddhita suffixes by Ashwini Deo (Deo, 2003) suggests that the semantic combinators ought to be classified according to a logic of descriptions, in an object-oriented fashion.

We shall discuss compounding in a specific section below, since it mixes morphology and syntax aspects.

## 5 The semantic side of syntax

We have a similar situation at the level of sentences. A sentence ought to be described as the composition of sign operations, whose semantic components form a dependency graph linking together the various words. Let us first consider the predicate of the sentence. It may be a finite verb form, in which case the stem formation is expressed by a *lakāra* indicating its tense and mood, seen as parameter of the corresponding action, indicated by a pair (preverb,root), plus possible parameters such as present class and secondary conjugation. Again this selection triggers a sign construction, consisting of a phonemic form paired with a paraphrase, with the action denoted as a sememe in a lexical entry of the corresponding verb. The choice of the verb entry involves possible disambiguation of the root (e.g. the choice between the two homonymic roots *dhāv*) and of the preverb (e.g. the choice between the privative and the intensive meanings of preverb *vi*-). Within the corresponding verb table, we may select a specific sememe, possibly expressed as a synset in the sense of WordNet. Finally, the finite verb form is specified by the conjugation parameters, i.e. number and person. The paraphrase of the form is a schema involving pronouns corresponding to the subject of the verb and its complements. Typically, in the active voice, the subject (possibly ellipsed) will denote the Agent, and an Object complement indicates the object/goal of the action in case the verb is transitive (which will have to be linked to an accusative phrasal sign when we fulfill the *kāraka* requirement). In the passive voice (*karmaṇi prayoga*) we have similarly two possible paraphrases, a passive paraphrase for transitive verbs, and an impersonal one for intransitive ones.

The verb itself may be ellipsed (implicit copula), and an appropriate paraphrase produced for the corresponding nominal sentences, with similar treatment. See (Kiparsky, 2009) for details. The precise formulation of the needed dependency structures is under investigation by Amba Kulkarni, who has defined a translation formalism for compiling *sūtra*s into valency requirements (*ākāṅkṣā*s) for the word forms of the sentence. She has used it to implement a dependency parser producing a dependency structure reflecting Pāṇinian constraints (Kulkarni, 2013; Kulkarni and Ramakrishnamacharyulu, 2013; Kulkarni et al., 2010). The corresponding conceptual graph is exactly what we need for our semantic combinators composition. Since the graph is presumed to be acyclic, we may express it as a tree, whose nodes are equivalence classes of the word forms (under the co-denotation equivalence that is needed to account for noun phrases combining adjectives and nouns in apposition and coordination) and whose arcs are relations corresponding to the semantic roles assignments. This tree (or more exactly directed acyclic graph) reflects the sentence-level structure of the signs operations.

Now what about the phonetic operations associated with this structure? They are actually separately specified, as the precedence ordering of the words of the sentence, that is more or less independent of the semantic operations (Staal, 1967; Gillon, 1995; Gillon, 1996). Thus the final sign construction representing the sequence is the list of words as nodes of the dependency structure. The production of the phonetic signal is obtained by composition, using external sandhi, of the phonetic realisations of the succeeding words, followed by final normalization by the sandhi rules from the final *tripādī* section of the *Aṣṭādhyāyī* (from 8.2.1 onwards). Alternatively, this final phonetic smoothing may be effected separately for every word of the sentence, since there is no spill-over of retroflexion rules across word boundaries (Goyal and Huet, 2013). These two strategies correspond to different, but equivalent, interpretations of the succession of *sūtra*s to be used to justify a Pāṇinian derivation (*prakriyā*).

We note *en passant* that a little bit of the semantic structure is needed in order to apply correctly sandhi, since forms in the dual number and vocatives have specific sandhi conventions. This further vindicates our integrated view of enunciation and sense in the notion of sign.

The semantic operation associated with the sentence structure consists of two parts. First, it is the canonical paraphrase associated with the semantic combinators tree (using pronouns for the sharing of the graph). Secondly, it is the optional list of addresses indicated by the vocative segments of the enunciation, which are independent of the dependency graph, as depending rather on the discourse structure (turn of speech, etc.). In the paraphrase, we have the choice of putting these addresses as some initial compound address (Ô *Guḍākeśa*, ...), or to insert these addresses at their precedence occurrence, if one wants to preserve more faithfully the sentence rhythm in the global paraphrase.

This dissociates clearly the meaning component (paraphrase of the dependency structure) from the syntax proper, reduced to a specific ordering of the words in the sentence. This permits to factor the sense out of the possibly dislocated utterances corresponding to the choice of the precedence of the words. Prose may be characterized by the consistency of the dependency graph with respect to the utterance, verified by non-crossing projections. Poetry may take advantage of the relative freeness of word order in order to choose enunciations consistent with the target meter, at the expense of possible dislocation. But dislocations may also be favored, for their "tortuosity" (*vakratā*) considered as an esthetic quality (*rasa*).

This view of sentence signs may be further expanded to actual interpretation of full texts, using concepts of discourse analysis (topic, theme, anaphora, etc.) A preliminary investigation of discourse analysis in Sanskrit is available in (Kulkarni and Das, 2012).

## 6 Compounds

### 6.1 Classification of compounds

Compounds are an unavoidable important feature of Sanskrit. Much has been said about their linguistic analysis (Gillon, 2007; Gillon, 1993; Gillon, 2002; Gillon, 2009; Kumar et al., 2010; Kulkarni and Kumar, 2013; Kulkarni and Kumar, 2011). The first misunderstanding to dispell is that compounds are not just phonemic strings. Compounds are structured items obtained by sign composition, and the resulting final enunciation (typically by glueing with external sandhi a number of bare stems to a final inflected form) forgets too much of its structure to be unambiguously paraphrased. Thus while *pītāmbaraḥ* is undeniably an exocentric compound (*bahuvrīhi*) glossed as "who has a yellow garment" (adjective itself reifiable in the noun "he who has a yellow garment", typically designating *Viṣṇu*), the accusative form *pītāmbaram* is ambiguous, since it might as well designate just a yellow garment as a *karmadhāraya*, now an autonomous noun phrase. This ambiguity is lifted in the traditional grammar, since compound formation is explained through abbreviation (*lopa*) of a proper noun or adjectival phrase (adverbial for the *avyayībhāva* family). It is the abstract semantic combinator that is the starting point of the derivation of the compound, and that unambiguously explains its meaning through a *laukika* paraphrase. From the relation between the components a proper case is selected, its corresponding inflected form is derived, only to be chopped off of its declension suffix (*lopa*) to obtain the final contracted enunciation (except in the rare cases of so-called *aluk* compounds, where the contraction does not take place, usually for proper names whose form was frozen in Vedic times). Actually, it should not be inferred that Pāṇini used this elision device by economy (*lāghava*), in order to deal with *aluk* compounds in the same way as the general case, in specifying that *lopa* is optional. What is most important is that the derivation ought to specify the semantic role, in order to obtain a non-ambiguous paraphrase expressing the meaning of the construction. This *kāraka*, or more generally its *sādhana* (for instance, the various shades of genitive meanings in the case of *ṣaṣṭhītatpuruṣa*s), has to be made explicit as a semantic combinator in the semantic component of the construct. Again we see one more justification of the grammar operating on the signs, and not just on their phonetic realisations.

Actually, this hints at a finer classification of *tatpuruṣa* compounds, since the assignment of case to *kāraka* involves choice, and thus we lose some information in keeping the case only. We could also distinguish the shades of meanings of *adhikaraṇa* in the case of *saptamītatpuruṣa*s, in order to distinguish e.g. time from space specification. In general, the relation (*bhāva*) between the compounded elements, specially when they are substantives, is of various natures: predication, comparison (*upamāna-upameya*), part-whole (*avayava-avayavī*), etc. A detailed semantic classification has been worked out by Prof K.V. Ramakrishnamacharyulu and collaborators in view of computational treatment.[2]

### 6.2 *Kṛt* suffixes as compound generators

This is the place to return to our discussion about *kṛdanta* formation in the case of an agentive suffix applied to a transitive root. For instance, rule 3.2.1 states *karmaṇyaṇ*, which expands by proper *anuvṛtti* to: "Affix *aṆ* occurs after a verbal root when the root co-occurs with a *pada* which denotes *karma* 'object'." (See e.g. (Sharma, 1987) vol III p 350–352). This rule concerns transitive verbs, and the expression of a primary derivative (*kṛdanta*) with an agentive meaning, conditioned on the concomitant production of the corresponding patient/goal (*karma*), typically as the left component of a compound word whose right component is the *kṛdanta*. It allows the construction of e.g. *kumbhakāraḥ* 'pot maker'. But here a problem appears in the control of the rules application. In order to justify the compound construction as the glueing of a *pūrvapada* (here *kumbhasya*) to the *uttarapada* (here *kāraḥ*) we must justify producing this genitive form (using rule 2.3.65) whose existence presupposes the *kṛdanta* which is in the process of being formed. This leads to a vicious circle (*itaretarāśrayadoṣa*). Furthermore, some hand-waving must be performed in order to justify classifying the compound as a *dvitīyā* (consistent with expressing its Object nature) rather than a *ṣaṣṭhī* (consistently with the genitive of its generating

[2] Ramakrishnamacharyulu, K. V., Kulkarni, A. P., Kulkarni, T., Kumar, A.: Guidelines for Tagging of Sanskrit Compounds. Private communication, 12.03.2012.

*pūrvapada*). As Pr. Rama Nath Sharma admits: "The derivation of *kumbhakāraḥ* is highly complex" (Sharma, 1987) Vol. 1 p. 179.

Such hairsplitting argumentation is typical of difficulties in trying to do two things at once, namely constructing a form and justifying its legal construction by grammar rules. These are two distinct aspects of using the grammar, that ought to be clearly separated and sequentialized. Our proposal is to separate the construction of the 'script' of the derivation (as a structure articulating the sign operations postulated as the locutor's communicative intention) from its verification, using *Aṣṭādhyāyī* rules. In this precise case, the compound construction will be directly predicated as some structure (Agent_*aṆ* Object Action) with Object the sign [*kumbha*/pot] and Action the sign [*kṛ*/making], eschewing the question of which case declension to apply to its first component, only to be deleted. At verification time, rule 3.2.1 will be applied to check that *kṛ* is indeed transitive, and that (recursively) the Object sign expression is valid. At execution time, proper rules will be applied to compute the *vṛddhi* form *kāra*, and produce the compound stem sign [*kumbhakāra*/he who makes a pot]. At sentence level, the role of the potter, assumed here to govern Nominative, will produce [*kumbhakāraḥ*/he who makes a pot] as a subject sign, etc.

We see clearly on this example the benefits of dissociating the sign elaboration from its later justification. This is similar to programs written in a Turing-complete programming language and analysed for type conformity by the type-checker of the compiler. The front-part of the compiler has two passes: abstract syntax construction (some combinatorial term over an operator algebra), followed by type-checking (usually some global computation on the term). In our proposal, operators operate on signs, abstract syntax consists of scripts, and type-checking corresponds to verifying whether they are Pāṇinian or not. The added value is that this verification may be global. We check easily the conformity of *kumbhakāra* since its -*kāra* component is postulated in a context where its accompanying object *kumbha* is known. In terms of attribute grammars, the Pāṇinian justification may be obtained by solving constraints on the global script, using not only synthesised attributes, but inherited attributes as well, leading to more sophisticated control.

### 6.3 Verbs as first-class citizens

There is another sense in which *kṛdanta*s may be considered as compound generators, when the primary form is derived not just from a root, but from a root modified by preverbs (*upasarga*). For instance, *pratikṛta* is the past participle of verb *pratikṛ*, obtained by affixing preverb *prati*- to root *kṛ*. From the point of view of morphology, it suffices to generate the past participle *kṛta* of the root, since the form *pratikṛta* may then be obtained as a *prādi* compound. This works for all derivatives of the root, since e.g. the preterit prefix augment (*a*-) applies to the root stem, and not to the 'verb stem'. Furthermore, the preverb has kept some autonomy in sandhi, witness the *ihehi* notorious example. Thus current parsers split the preverbs sequence independently from the root forms, and participles such as *pratikṛta* are analysed as *prādi* compounds. This way the morphological banks may be limited to finite forms and *kṛdanta*s of roots.

This position is not satisfactory on the semantics side. Participles (and more generally *kṛt* constructions) are associated with generic aspects of the action denoted by the verb, whereas preverbs often acquire a specific meaning at the contact of the root, yielding *sui generis* actions. Thus the semantic combinators do not commute, and the *kṛdanta* of a verb with a preverb is usually more precise than using the preverb meaning as a modifier of the root *kṛdanta* meaning. We shall thus prefer keeping the association (preverb,root) as a 'first class citizen', lexicalized with its particular shades of meaning, and interpreting the *kṛt* constructions as direct primary derivatives, rather than *prādi* compounds.

Two facts come in support of this view of verbs (roots prefixed by preverbs) as first-class citizens. First, the fact that many of the *sūtra*s give conditions on the preverbs allowed for the relevant form. Thus morphology generation involves preverbs after all. The second hint that preverbs 'stick' closely to their root is that the sandhi joining them generally involves retroflexion, in opposition to the external sandhi glueing *pada*s.

Actually, this suggests generating all finite and *kṛdanta* formations for verbs given with their attested preverbs. After all, the number of such forms is finite, and sharing can contract their lexicon without much increase with respect to just storing root forms, if appropriate data structures are used.

This does not mean that *prādi* compounds should be altogether discarded. The above discussion concerns the so-called *gati* compounds, where the preverb is co-referential with the verbal action. We still need *prādi* compounds in cases of non-compositionality, when the prepositional meaning of the particle differs from its preverb usage. For instance, we shall distinguish the primary derivative *nirvācya*, passive future participle of the causative of verb *nirvac* "to be explained" from the homophonic *prādi* compound *nis-vācya* meaning "irreproachable" or "that should not be said". Even though both stems derive from the same root *vac*, the prefix "nis-" is affixed to the root in the co-referential sense of "toward the outside" to form *nirvac*, whereas in it affixed to *kṛdanta vācya* "speakable" in the negative sense of "unspeakable". It is clear on this example, where senses are opposite, that we must identify the two constructions with distinct sign combinators.

### 6.4 Non-compositional compounds

Finally, let us say a few words about non-compositional (*asamartha*) compounds. These compounds are anomalous, in that their semantic interpretation involves external dependencies, not just at their periphery like for exocentric *bahuvrīhi*s. A typical example is *ripumānī bhrātuḥ* "he who thinks to be an enemy of his brother". An investigation by linguist Brendan Gillon (Gillon, 2007) reveals that such compounds are more often attested in the corpus as one would believe. Such compounds pose a real problem to the representation of the dependency structure of sentences, which is now pointing inside a compound, necessitating a non-compositional interpretation. This significantly complicates the model of sign operators, which must now operate in a context. On the other hand, this contextual handling (technically $\lambda$-calculus representations for continuations) is needed at the discourse structure, for instance for anaphora/cataphora treatment.

Another consideration is that the *asamartha* character depends on the granularity of the underlying concept calculus. Whether *gurukula* involves linking *guru* to its surrounding *śiṣya* to identify the family home where the teaching occurs, or whether it should be considered as a frozen concept of "traditional school" used generically is open to debate. This suggests that some compounds should just be by-passed by the semantic interpretation as idiomatic notions, rather than being analysed compositionally. Thus *nṛpa* should be just a king, *pādapa* should be just a tree, etc.

## 7 Revisiting the programming analogy

To return to the program analogy, the script is the program, its verification is type-checking, insuring consistency. Once this type-checking is effected (and without circularity now), the script is a *bona fide* program. Now executing the program is operating on the sign operators it is constituted of, until the final sign is produced, yielding both the enunciation and its meaning (as a paraphrase). A bottom-up traversal yields the resulting sign, both in its signifiant, the full sentence, and its signifié, as a paraphrase.

Furthermore, this computation may yield, as its trace, the complete sequence of *sūtra*s necessary for its derivation, provided enough information is explicit in the script. However, note that we have an extra degree of freedom, since the script may be executed in a bottom-up fashion in various orders, leading to possibly many equivalent *prakriyā*s. For instance, the words in the sentence may be elaborated in a succession order different from the word order at the sentence level. This is an important abstraction principle.

Further abstractions are possible. Typically, words may be lexicalized, who do not need to state all details of their elaboration from roots. If the lexicon has a careful organisation in terms of sememes labeled with their sign combinator, and if it is verified at lexicon construction time that these signs are indeed derivable in the grammar, now we may safely incorporate lexicon access as abstract nodes in the script structure. Thus for instance -*kāra* could be lexicalized with a sign combinator usable as ifc (*uttarapada*) of any compound whose iic (*pūrvapada*) component is labeled Object. This sign combinator would be something like the schema -*kāra*/iic-maker. We may abstract further if the lexicon has the proper vocabulary coverage. For instance, we may abstract the pot-maker compound as [*kumbhakāra*/potter] if the stem *kumbhakāra* is properly lexicalized. More generally, we should reserve morphological analysis to words whose meaning is compositional (*yaugika*), and allow direct lexicon access for words whose meaning is conventional (*rūḍha*).

We need also to make explicit proper names, which should not be analysed, such as *Rāmāyaṇa*, which is a frozen form with retroflexion, or worse *Yudhiṣṭhira*, which not only incurs retroflexion, but is an *aluk* compound, with its *pūrvapada* a locative form. We shall not elaborate further on the computational treatment of Sanskrit proper names, which is a complex matter.

We can actually use the lexicon as an extended *gaṇapatha*, furnishing attested forms without having to postulate artificial roots. This organisation of a grammatical lexicon is now a well-established feature of modern computational linguistics platforms. For instance, the Tree Adjoining Grammars (TAG) concept of Aravind Joshi gave rise to parsers exploiting specialized lexicons documenting the syntactic combinatorial effect (e.g. tree adjunction) associated with lexical items.

At the other extreme, we might replace in *śāstra* argumentations lexical occurrences of technical notions by their precise paraphrase as a *navyanyāya* definition. Thus we have a great flexibility in considering the script as a formal object, subject to expansion and contraction as the need arises.

## 8 Adapting the lexicon structure

The homophony/polysemy of various particles is a problematic matter. For instance, prefix *vi-* may be used as an intensifier or a negator, leading to quasi-opposite meanings. Obviously we shall have to distinguish two distinct particles in the two homophones of *vibuddha* meaning respectively "awake" and "unconscious". Also, the privative particle *a-/an-* has several meanings which must be distinguished.

Similarly the co-denotative nature or not of the prefix may lead to distinct homophones, as we saw above with *nirvācya*. Even though these homophones come from the same root, their distinct morphological construction demands separate lexical entries. More so if distinct roots are involved, like *vimāna*$_1$, primary derivative in the agentive sense of *vimā* "who traverses" (and by extension an aerial vehicle), to be distinguished from the *prādi* compound *vimāna*$_2$ 'disrespect'. A specially striking example is *samāna*, that admits of at least 4 analyses, leading to 4 distinct meanings (digestion; similar; honored, proud; of the same measure) (Raja, 1969).

We have similar problems for distinguishing descriptive (*tatpuruṣa*) meanings of compound stems from their exocentric (*bahuvrīhi*) ones. Although here the accent is different, and if our signifiants retain their accent, as specified by Pāṇini, the two lexical entries may be readily distinguished if lexemes are presented with phonetic features such as accent. In any case, "raising to *bahuvrīhi*" will have to be an explicit sign combinator (noted -B in Gillon's notation).

A common polysemy problem concerns the substantive stems in *-ana*. They may typically denote action words, or states, like neuter *nejana* (cleaning). But they may also denote adjectival agents in all genders. For instance, *pācana* may be a cook as well as cooking. Here we have distinct *kṛt* suffixes for the two meanings, corresponding to distinct roles (*sādhana*), namely Agent and Action. We must distinguish in entry *pācana* the two sections pertaining to these different roles. For each role, we may further ramify in possibly multiple *kṛt* suffixes leading to semantic shades of meaning (such as frequent agent, enjoying agent, professional agent, etc.). Finally, each specific *kṛt* section may refer to the Pāṇinian *sūtra* governing its introduction for this specific sememe.

We have started making use of these ideas to the adaptation of the Sanskrit Heritage Dictionary as a Pāṇinian-informed repository. We structure nominal entries as explicit *kṛt* or *taddhita* formations[3]. When known, the corresponding production *sūtra* is referenced. In the case of *kṛt* suffixes, we adopt the following policy. Participial, Infinitive and Absolutive forms are systematically produced by morphology generation for most roots. Nominal formations are not generally productive, and must therefore be lexicalized. In case several formations from a giving root lead to the same stem, we use the stem as unique lexeme, informed with the various *sādhana*s. Each *sādhana* is again partitioned in the various possible *kṛt* suffixes. Then the various sememes are grouped by successive lists of senses, each sense being characterized by a synset in the sense of WordNet. We intend to use this "Pāṇinianized" lexicon as a tool for testing the ideas of the present paper.

[3] The Hindi version of Apte's dictionary is similarly informed.

A more ambitious pursuit would be to actually link sememes to a precise WordNet repository. Here there is a choice. We could use a precise alignment to the Monier-Williams dictionary, the most complete Sanskrit dictionary publically available under a computable XML format. This alignment, composed with the reference English WordNet, would give a synset index available as some kind of multilingual semantic reference for atomic Sanskrit signs. The semantic combinators could be expressed as paraphrases in a target language, allowing a sort of literal translation in any language having a WordNet interface. The Sanskrit WordNet under development at IITB would be another alternative. Its implementation, consistent with the Hindi WordNet, would provide a finer discrimination for Indian specific notions.

A prerequisite of this effort would be insure that scripts make explicit the nominal or adjectival role of the *subanta* stems, either by requiring script editors to demand this part of speech information at lexicon selection, or else by post-processing scripts to reveal this information, after inspecting the sentence structure. It is to be noted that Gillon's phrase structure tree combinators distinguish adjectives from nominals. On the other hand, more flexibility could allow an easier handling of dependency structures, avoiding explicit "empty nodes" reifying an adjective into the ellipsed topic that it qualifies, a common construction. Here what matters is the interface between sentence and discourse - if the discourse structure is available, the ellipse turns into an anaphora reference, and a more precise semantic analysis may be expressed.

If the sememe structure of the lexicon is fine enough, the information about adjectival versus nominal use will be available. We may even refine this idea in using the lexicon as a repository of proper names, if it encompasses an encyclopedic structure. This would endow our scripts with named-entity specification, and allow us to distinguish scripts containing forms of word *kṛṣṇa* as referring to some black object, to *Kṛṣṇa* as Hero/God, or to whoever happens in the current context to be called *Kṛṣṇa*.

## 9 A roadmap for a concrete implementation of these design principles

Precise design of the script algebra is a non-trivial task, since there are issues of granularity and of lexicon interface. Allowing it at the three levels word-sentence-discourse involves properly modeling anaphora, ellipsis, indirect speech, etc.

The design of the script formalism should go in parallel with the development of a structured script editor, amenable to consistent manipulation of dependency-based discourse structures. Schemas over this formal structure (presumably defined in some kind of typed $\lambda$-calculus) ought to provide the proper notion of paraphrase for each elementary sign combinator. The editor would allow a Sanskrit locutor to specify his communicative intention in terms of the script semantic combinators.

Many degrees of freedom exist in the design of the script editor. The editor may work bottom-up, and thus cross the word/sentence level by requesting the user to assign a *kāraka* to a *pada* given with case inflexion. We could also imagine working top-down, first specifying semantic roles in the sentence before choosing lexical elements to populate them. This last scenario is closer to the way in which locutor intentions are explained in analysis, but it may be harder to design - a tree is built inductively in a bottom-up way, whereas its top-down construction needs meta-notation for context management.

Another degree of freedom concerns the specification of the words precedence in the sentence, which may be delayed to a certain extent. Some point of view may be preferred over another one for specific applications (e.g. analysing existing corpus versus teaching tool guiding students in Sanskrit composition).

The next software component would be a Pāṇinian verifier, that will check that every script constructor is validated by a proper *sūtra* sequence. This is enormous work, since it involves verifying the control meta-rules for *sūtra* context management, but it could benefit of all the experience accumulated by the designers of Pāṇinian simulators.

The last component would be a script evaluator, that will reduce a Pāṇinian script to its final Sanskrit sign [enunciation/meaning], with 'meaning' a non-ambiguous paraphrase of 'enunciation', guaranteed to be a Pāṇinian Sanskrit utterance. This evaluator is basically routine algorithmics. It would be interesting to make it parametric with a notion of evaluation strategy, notably at the level of the order of the words in the sentence. It could then be coupled with a meter recognizer, that could enforce/verify poetic metrics.

Actually, the script evaluator is independent of the grammar verification, opening the possibility to construct, manipulate and execute un-Pāṇinian Sanskrit scripts. This would be useful to philologists working on e.g. epic Sanskrit with ungrammatical forms, or to proponents of contemporary usage of Sanskrit, with neologisms, technical vocabulary, and foreign words borrowings.

Many other formal manipulations could be envisioned. For instance, a prose translator, that would untangle dependencies in order to produce a non-dislocated phrase-structure representation in the sense of Gillon. This would facilitate interfacing scripts with the dependency structures produced by current parser technology.

## Conclusion

We have presented a novel proposal for a computational model of the *Aṣṭādhyāyī*. It relies on the notion of *sign*, associating a phonetic production and its sense, formalized as a paraphrase issued from its morphology. This applies at 2 levels, for words and for sentences. Each grammatical construction is modeled as a sign combinator, where a linguistic *sign* is a pair [enunciation,meaning](*śabdārthau*). The locutor intention (*tātparya*) is expressed by a *script*, formal object over a sign combinators language. An important degree of freedom in this design is that a Sanskrit sign may not be conformant to Pāṇinian constraints.

The script may be verified by a Pāṇinian checker, that admits correct grammatical constructions and rejects others. A Pāṇinian script represents a *bona fide* computational object, similar to a program operating on Sanskrit signs. Execution of the script yields both the correct enunciation, and a non-ambiguous paraphrase of its meaning. Lexicalization allows a level of abstraction, possibly leaving to verification only the sentence structure.

These ideas are under experimentation with the Sanskrit Heritage platform, notably an on-going marking of primary derivatives by sign combinators. The design of the sign combinator formalism fit to account for the variety of Pāṇinian specifications is an interesting challenge. It should benefit from the current efforts in Sanskrit parsers using dependency grammar formalisms.

### Acknowledgments

A preliminary version of these ideas was presented as a talk at the Pondicherry Workshop on Pāṇini and the Pāṇinians organised by Jan Houben and François Grimal in October 2014. The author wishes to thank the numerous Sanskrit scholars who patiently explained rudiments of *vyākaraṇa* to him, in order to remove his Western Linguistics biases and make him understand the true semantic nature of *Aṣṭādhyāyī*.

## References

Saroja Bhate and Subash Kak. 1993. Panini's grammar and computer science. *Annals of the Bhandarkar Oriental Research Institute (72) 7*, pages 79–94.

Saroja Bhate. 1989. *Pāṇini's taddhita rules*. Center of Advanced Study in Sanskrit, University of Poona, Pune.

George Cardona. 1965. On translating and formalizing Pāṇinian rules. *Journal of Oriental Institute, Baroda, vol. 14*, pages 306–314.

George Cardona. 1974. On Pāṇini's metalinguistic use of cases. In *Charudeva Shastri Felicitation Volume*, pages 305–326. Charudeva Shastri Felicitation Committee.

George Cardona. 1976. *Pāṇini, A Survey of Research*. Mouton, The Hague.

Ferdinand de Saussure. 1916. *Cours de linguistique générale*. Payot.

Ashwini Deo. 2003. Derivational morphology in inheritance-based lexica: Insights from Pāṇini. *Lingua 117*.

B. Dīkṣita, S.C. Vasu, and B. Das Basu. 1905. *The Siddhānta kaumudi of Bhattoji Dikshita*. Number v. 3 in The Siddhanta Kaumudi of Bhattoji Dikshita. The Panini office.

Pierre Filliozat. 1975. *Le Mahābhāṣya de Patañjali, avec le Pradīpa de Kaiyaṭa et l'Uddyota de Nāgeśa*. Number 54,1-5 in Publications de l'Institut Français d'Indologie. Pondichéry.

Brendan S. Gillon. 1993. Bartṛhari's solution to the problem of asamartha compounds. *Études Asiatiques/Asiatiche Studien*, 47,1:117–133.

Brendan S. Gillon. 1995. Autonomy of word formation: evidence from Classical Sanskrit. *Indian Linguistics, 56 (1-4)*, pages 15–52.

Brendan S. Gillon. 1996. Word order in Classical Sanskrit. *Indian Linguistics*, 57,1:1–35.

Brendan S. Gillon, 2002. *Indian linguistic studies: Festschrift in honour of George Cardona; Eds. Madhav M. Deshpande and Peter E. Hook*, chapter Bartṛhari's rule for unexpressed kārakas: The problem of control in Classical Sanskrit. Motilal Banarsidass, Delhi.

Brendan S. Gillon. 2007. Exocentric (bahuvrīhi) compounds in classical Sanskrit. In Gérard Huet and Amba Kulkarni, editors, *Proceedings, First International Symposium on Sanskrit Computational Linguistics*, pages 1–12.

Brendan S. Gillon. 2009. Tagging classical Sanskrit compounds. In Amba Kulkarni and Gérard Huet, editors, *Sanskrit Computational Linguistics 3*, pages 98–105. Springer-Verlag LNAI 5406.

Pawan Goyal and Gérard Huet. 2013. Completeness analysis of a Sanskrit reader. In Malhar Kulkarni and Chaitali Dangarikar, editors, *Recent Researches in Sanskrit Computational Linguistics (Proceedings, 5th International Symposium on Sanskrit Computational Linguistics)*, pages 130–171. D.K. Printworld.

Pawan Goyal, Amba Kulkarni, and Laxmidhar Behera. 2009. Computer simulation of *aṣṭadhyāyī*: Some insights. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 139–161. Springer-Verlag LNAI 5402.

F. Grimal, V. Venkataraja Sarma, V. Srivatsakacharya, and S. Lakshminarasimham. 2006. *La grammaire Paninéenne par ses exemples*. Rashtriya Sanskrit Vidyapeetha, EFEO Pondicherry and IFP.

Jan E.M. Houben. 1999. 'Meaning statements' in Pāṇini's grammar: on the purpose and context of the Aṣṭadhyāyī. *Studien zur Indologie und Iranistik, band 22*.

Jan E.M. Houben, 2002. *Indian linguistic studies: Festschrift in honour of George Cardona; Eds. Madhav M. Deshpande and Peter E. Hook*, chapter The sequencelessness of the signifier in Bartṛhari's theory of language. Motilal Banarsidass, Delhi.

Gérard Huet. 2005. A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger. *J. Functional Programming*, 15,4:573–614.

Malcolm Hyman. 2009. From Pāṇinian sandhi to finite state calculus. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 253–265. Springer-Verlag LNAI 5402.

Peter Ingerman. 1967. Panini-Backus form suggested. *Communications of the ACM 10 (3)*.

S.D. Joshi and J.A.F. Roodbergen. 1990. *Patañjali's Vyākaraṇa-Mahābhāṣya Sthānivadbhāvāhnika: introduction, text, translation and notes*. Number v. 1 in Research Unit series. Bhandarkar Oriental Research Institute.

S.D. Joshi and J.A.F. Roodbergen. 2004. *The Aṣṭādhyāyī of Pāṇini with Translation and Explanatory Notes*. Number v. 11 in The Aṣṭādhyāyī of Pāṇini. Sāhitya Akādemī.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20,3:331–378.

Paul Kiparsky and J. F. Staal. 1969. Syntactic and semantic relations in Pāṇini. *Foundations of Language*, 5:83–117.

Paul Kiparsky. 1980. *Pāṇini as a Variationist*. Center of Advanced Study in Sanskrit, University of Poona, Pune.

Paul Kiparsky. 2009. On the architecture of Pāṇini's grammar. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*. Springer-Verlag LNAI 5402.

K. Koskenniemi. 1984. A general computational model for word-form recognition and production. In *10th International Conference on Computational Linguistics*.

Amba Kulkarni and Monali Das. 2012. Discourse analysis of Sanskrit texts. In *Proceedings of the Workshop on Advances in Discourse Analysis and its Computational Aspects*, pages 1–16, Mumbai, India, December. The COLING 2012 Organizing Committee.

Amba Kulkarni and Anil Kumar. 2011. Statistical constituency parser for sanskrit compounds. In *Proceedings of the 8th International Conference on NLP 2011*. Macmillan Advanced Research Series, Macmillan Publishers India Ltd.

Amba Kulkarni and Anil Kumar. 2013. Clues from Aṣṭādhyāyī for compound type identification. In Malhar Kulkarni and Chaitali Dangarikar, editors, *Proceedings of the 5th International Sanskrit Computational Linguistics Symposium*. D K Printworld.

Amba Kulkarni and K V Ramakrishnamacharyulu. 2013. Parsing Sanskrit texts: Some relation specific issues. In Malhar Kulkarni and Chaitali Dangarikar, editors, *Recent Researches in Sanskrit Computational Linguistics (Proceedings, 5th International Symposium on Sanskrit Computational Linguistics)*, pages 130–171. D.K. Printworld.

Amba Kulkarni, Sheetal Pokar, and Devanand Shukl. 2010. Designing a constraint based parser for Sanskrit. In G N Jha, editor, *Proceedings of the 4th International Sanskrit Computational Linguistics Symposium*. Springer-Verlag LNAI 6465.

Amba Kulkarni, Preeti Shukla, Pavankumar Satuluri, and Devanand Shukl. 2015. How free is 'free' word order in Sanskrit? In Peter Scharf, editor, *Sanskrit syntax: Selected papers presented at the seminar on Sanskrit syntax and discourse structures, Paris, June 2013*. The Sanskrit Library, Providence, R.I.

Amba Kulkarni. 2013. A deterministic dependency parser with dynamic programming for Sanskrit. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 157–166, Prague, Czech Republic, August. Charles University in Prague, Matfyzpress, Prague, Czech Republic.

Anil Kumar, Vipul Mittal, and Amba Kulkarni. 2010. Sanskrit compound processor. In G N Jha, editor, *Proceedings of the 4th International Sanskrit Computational Linguistics Symposium*. Springer-Verlag LNAI 6465.

Anand Mishra. 2009. Modelling the grammatical circle of the Pāṇinian system of Sanskrit grammar. In Amba Kulkarni and Gérard Huet, editors, *Proceedings, Third International Symposium on Sanskrit Computational Linguistics*, volume LNAI 5406. Springer.

Anand Mishra. 2010. Modelling Aṣṭādhyāyī: An approach based on the methodology of ancillary disciplines (Vedāṅgas). In Girish Nath Jha, editor, *Proceedings, Fourth International Symposium on Sanskrit Computational Linguistics*, volume LNAI 6465. Springer.

K. Kunjunni Raja. 1969. *Indian Theories of Meaning*. The Adyar Library and Research Center, Chennai.

Peter Scharf. 2009. Modeling Pāṇinian grammar. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 95–126. Springer-Verlag LNAI 5402.

Hartmut Scharfe. 1971. *Pāṇini's metalanguage*. v89, Memoirs of the American Philosophical Society, Philadelphia.

A. Sharma, K. Deshpande, and D.G. Padhye. 2008. *Kāśikā: A Commentary on Pāṇini's Grammar*. Sanskrit Academy series. Sanskrit Academy, Osmania Universtiy.

Rama Nath Sharma. 1987. *The Aṣṭādhyāyī of Pāṇini*. Munshiram Manoharlal, Delhi.

J. F. Staal. 1967. *Word Order in Sanskrit and Universal Grammar*. Reidel, Dordrecht.

Lucien Tesnière. 1959. *Éléments de Syntaxe Structurale*. Klincksieck, Paris.

Paul Thieme. 1983. Meaning and form of the 'grammar' of Pāṇini. *Studien zur Indologie und Iranistik Heft 8/9 1982/3*, pages 3–34.