Chapter 1
Introduction

The study of Aṣṭādhyāyī can be classified into three broad areas of academic research:

1. Analysis of the grammatical corpus in order to understand its organization and functioning,
2. formalization of the grammatical system, and
3. its computer implementation or automation.[1]

The present work deals with the latter two areas, namely, formalization and computer implementation of Aṣṭādhyāyī. It seeks to study the content and processes of the Pāṇinian system of Sanskrit grammar and *re-present* them in terms of logical relations and operations. A formal representation is attempted in order to facilitate an examination of the underlying grammatical structures. It also enables an implementation of the grammatical processes on computer.

1.1 Earlier research

In the past few decades there has been an increased interest in studying the Aṣṭādhyāyī from a formal perspective. Scholars like Vidya Niwas Misra (1964, 1966), M. D. Pandit (1966, 1974), Frits Staal (1965, 1966) and Paul Kiparsky

[1] According to Frits Staal (1966 p. 209): "If we distinguish three stages in the study of Aṣṭādhyāyī as a generative device, it may be held that the first stage, that of analysis, has been dealt with successfully by Indian commentators since Patañjali and by Western scholars of the last two centuries; however, this task is by no means completed. The second stage, that of formalization, has perhaps just begun to receive attention; it depends on analysis, but is not determined by it. The third stage, that of automation, itself depending on formalization, is not determined by formalization; it may not even be effectively realizable."

(1969) published research papers and monographs showing that certain formal features of mathematics and modern linguistic theories like context-sensitive rules or elements of generative transformational grammar are already present in it. These studies further supported the initial fascination for the Aṣṭādhyāyī as "one of the greatest monuments of human intelligence" and "an indispensable model for the description of languages" (Bloomfield 1929 p. 268). In the year 1985 Rick Briggs, a NASA scientist, published a paper on "Knowledge Representation in Sanskrit and Artificial Intelligence" in which he compared the system of *kāraka*s with representational techniques in Artificial Intelligence and posited that:

> Among the accomplishments of the grammarians can be reckoned a method for paraphrasing Sanskrit in a manner that is identical not only in essence but in form with current work in Artificial Intelligence.[2]

This statement is illustrative of the aim of extracting the techniques of representation in the Pāṇinian grammar that can be fruitfully employed for computational processing. Following this, a "National Conference on Knowledge Representation and Inference in Sanskrit" was organized in Bangalore in December 1986, "to extract this hidden 'algorithm' of automatic semantic parsing from the Sanskrit pandits" (Briggs 1987 p. 99). A group of scholars from the Indian Institute of Technology, Kanpur undertook projects incorporating Pāṇinian perspectives, especially the *kāraka*-system, with modern techniques of Natural Language Processing (NLP). Their aim was to develop a machine translation tool for English and Hindi based on insights gained by the Aṣṭādhyāyī of Pāṇini (Bharati 1994). The work initiated is followed by the "AnusAraka" Language Resource Development project. This is still an ongoing project, and once completed, should "allow users to access text in any Indian language, after translation from the source language (i.e. English or any other regional Indian language)".[3]

The nature of the above efforts has been to utilize some of the insights from the Pāṇinian grammar and apply them to the standard techniques of Natural Language Processing. The next step comes from researchers working in the field of computational linguistics. A general opinion which is often articulated here is that Sanskrit is one of the most suitable languages for computers. This is normally grounded on the assumption that it is a well-structured language which in turn is justified on the basis of the algebraic rules of its grammar. Some scholars opine that Sanskrit, being a perfect language, with a grammar like Aṣṭādhyāyī, comes closer to a computer

---

[2] Briggs' analysis is based on the Vaiyākaraṇasiddhāntamañjūṣā of Nāgeśa Bhaṭṭa (1730-1810 C.E.). See (Briggs 1985 p. 32-34).

[3] The partner institutions of this project are: Chinmaya International Foundation (CIF) Shodha Sansthan, Kerala; Language Technologies Research Centre, IIT Hyderabad; Department of Sanskrit Studies, Hyderabad University. For more information, see the project website: `http://ltrc.iiit.ac.in/~anusaaraka/` (accessed on 24.10.2015).

language and in future even computer programs could be written in Sanskrit. A summary of the approaches followed by the ongoing research projects on computerization of the Aṣṭādhyāyī, however, shows that as yet there are no finished automated systems or programs that implement the whole corpus of Aṣṭādhyāyī.

A first effort in this regard is the creation of an electronic version of the corpus of Aṣṭādhyāyī, which was prepared by Dr. Shivamurthy Swamiji of Sri Taralabalu Jagadguru Brihanmath, Sirigere, Karnataka. He calls it Gaṇakāṣṭādhyāyī meaning "computer software on Aṣṭādhyāyī".[4] It contains the Sūtrapāṭha, Padapāṭha, *anuvṛtti*s, *vṛtti* from Siddhāntakaumudī and Laghusiddhāntakaumudī (incomplete), French translation by Louis Renou as well as inflectional tables for nominal and verbal stems, including step-by-step analysis of Pāṇini's *sūtra*s, applied to produce different forms of nominal stems. Shivamurthy Swamiji is also developing a rule based application of the Pāṇinian derivational process.[5]

Another database of examples (*udāharaṇa*) found in the four major commentaries of the Aṣṭādhyāyī—namely the Mahābhāṣya, Kāśikāvṛtti, Bhāṣāvṛtti and Siddhāntakaumudī—is prepared by the French Institute of Pondicherry.[6] These are published in printed form as well as CD-ROM version including books on collection of examples (*udāharaṇa-samāhāra*), on compounds (*samāsa prakaraṇam*) and on verb inflections (*tiṅanta prakaraṇam*) (Grimal 2005, 2006, 2006a and 2010).

A digital edition of the Aṣṭādhyāyī is being prepared by Wiebke Petersen under the project: "*Pratyāhāra*s or features? A qualitative analysis of phonological descriptive techniques—a comparison of Pāṇini's *pratyāhāra*s and phonological features".[7] Attempts to implement the content and processes of the Aṣṭādhyāyī are relatively recent and only a few in number. Most of them base themselves upon the research and publications in the area of formalization of the Pāṇinian grammar. It is imperative, therefore, to first look into the outcome of the investigations in this field.

---

[4] Information accessed from `http://www.taralabalu.org/panini/` on 02.11.2015.

[5] During his visit to Heidelberg on 17.05.2013 he showed me his application for the declension of nominal stems which he hopes to finish in near future. He told me that he is attempting to follow the exact process of Aṣṭādhyāyī, although I did not had the opportunity to look into the program codes. Thus far, there is no publication on the manner in which it is implemented.

[6] For more information about the ongoing project, see the project website (accessed on 09.11.2015): `http://www.ifpindia.org/Paninian-Grammar-through-its-Examples.html`

[7] More information at the website: `http://panini.phil.hhu.de/?section=home` (accessed: 22.02.2016).

Apart from a few early publications that explored the mathematical aspects of Pāṇini,[8] the tone of the research towards formalization of the Aṣṭādhyāyī was set by developments in the generative grammar approach of Noam Chomsky in late fifties and early sixties. Chomsky declared Pāṇini's grammar to be the first and earliest version of a generative grammar.[9] The idea of a formal grammar of language that can generate an infinite number of utterances with a limited set of grammatical rules and a finite set of terms, evoked a close parallel with the Aṣṭādhyāyī. Accordingly, some Pāṇinian experts published papers with the prime aim of comparing and showing the presence of Chomskyan findings in the grammatical system of Pāṇini. In a paper written in 1965, Murray Fowler attempted to test whether Pāṇini's rules can be ordered in a manner so that they can be implemented through a Finite State Automaton (Fowler 1965 p. 44-47). This corresponds to the Type-3 or regular grammars in the Chomsky hierarchy.[10] Frits Staal promptly corrected this assumption in a brief communication in 1966, and showed that the way Pāṇini's rules are conceived and organized, it would not be possible to equate them with a regular grammar (Staal 1966 p. 206-209).

Staal further showed parallels with the Type-1 or context-sensitive grammars and certain phonetic rules for replacements of sounds in the Pāṇinian grammar. In the year 1965, he published an article on the "Context-sensitive rules in Pāṇini" (Staal 1965). He selected rules from the sixth chapter of the Aṣṭādhyāyī, mainly from 6.1.71 to 6.1.109. The rule *iko yaṇaci*[11] is a typical example. He took the terminology from Chomsky[12] and described this phenomenon using the representation

$$a[b \rightarrow c]d$$

[8] These include publications by Misra (1964 p. 157-178) and Pandit (1966, 1974 p. 179-192). These are, however, mostly of the nature of detecting some mathematical similarities. Among the early publications are the articles of Klaus Mylius (1980 p. 233-248) on the application of mathematical methods in the Vedic research which discusses mostly the statistical methods as also Madhav Deshpande (1992 p. 15-27) comparing the Pāṇinian features with developments in computational linguistics.

[9] The classical work of Chomsky in this regard is his book "Syntactic structures" (Chomsky 1957). Chomsky e.g. speaking at the Asiatic Society of India, Kolkata on 22.11.2001 tells that "the first generative grammar in the modern sense was Pāṇini's grammar" (Chattopadhyay 2001 p. 18).

[10] The grammars of a formal language are put in a hierarchy called the Chomsky hierarchy. The Type-0 corresponds to unrestricted grammars, Type-1 to context-sensitive grammars, Type-2 to context-free grammars and Type-3 to regular grammars. Chomsky hierarchy plays an important role in the area of formal languages which have special application in computer science, see (Chomsky 1956 p. 113-124, 1959 p. 137-167 ; Chomsky and Shützenberger 1963 p. 118-161).

[11] इको यणचि ॥६.१.७७॥ ▶ i*k* is replaced by ya*ṇ* if it is followed by a*c*.

[12] Staal provides the reference: (Chomsky 1963 p. 294).

where a is left context, d is right context, b is *sthānin* or to be replaced and c is *ādeśa* or replacement. In other words b is replaced by c.[13] He showed that Pāṇinian meta-language can even represent the process of substitution for more than one phoneme in a collective manner. This is demonstrated by the convention for respective correspondence of two lists of equal cardinality stated by the rule: *yathāsaṃkhyamanudeśaḥ samānām*[14]. This would be equivalent to the following representation:

$$a[b_1 \dots b_n \rightarrow c_1 \dots c_n]d$$

Here, b_1 is replaced by c_1, b_2 is replaced by c_2 etc. Further, he notes that if the contexts remain the same, then they need not be repeated every time and the idea of *anuvṛtti* (carrying over to subsequent rules) is applied in the grammatical corpus to present them in a more succint manner. Thus,

$$\begin{array}{l} a_1[b_1 \rightarrow c_1]d_1 \\ a_1[b_2 \rightarrow c_2]d_2 \end{array}$$

can be represented in a more concise manner as follows:

$$\begin{array}{r} a_1[b_1 \rightarrow c_1]d_1 \\ [b_2 \rightarrow c_2]d_2 \end{array}$$

Staal extended the comparison beyond the phonetic rules and published a paper in which he showed that the methods of generative grammar are similar to the syntax of nominal compounds in Sanskrit (Staal 1966a p. 198).

The comparison and motivation from the generative grammar was extended to the syntactic and semantic relations in Pāṇini in a paper published jointly by Paul Kiparsky and Frits Staal in 1969. In this paper, the authors proposed that Pāṇini's grammar is a system of rules for converting semantic representations of sentences (concepts like "agent", "goal", "location") into phonetic representations (case endings, verbal affixes etc.). This is achieved via two intermediate levels which may be respectively compared with the levels of deep (underlying) structure and surface structure in a generative grammar. The deep level corresponds to the level of *kāraka*-relations such as "(underlying) subject", "(underlying) object" and the surface level represents morphological categories like nominal cases, derivational affixes etc. (Kiparsky and Staal 1969 p. 84). While carrying out the comparisons with the generative grammar, they pointed out that there are essential differences as well, especially in the manner in which rules are ordered and organized in the Aṣṭādhyāyī and the way constituent structures are used (Kiparsky and Staal 1969 p. 105-106).

[13] The use of arrow here is different than the general notation, according to which $b \rightarrow c$ means: c is replaced by b.

[14] यथासंख्यमनुदेशः समानाम् ॥१.३.१०॥ ▶ respective assignment for equal number of elements.

The authors successively worked-out and improved this model, and the actual version is stated by Kiparsky in a paper published in 2009 (Kiparsky 2009 p. 35-37).[15]

The generative approach started by Staal's comparison of context-sensitive rules and extended and developed by other scholars like Kiparsky—and to some extent acknowledged by experts like Johannes Bronkhorst (1979 p. 146-157), S. D. Joshi and J. A. F. Roodbergen[16]—had a far reaching impact on the attempts by later scholars aiming for computerization of the Aṣṭādhyāyī. This will be evident from the following summary of these efforts.

In the year 1993, Saroja Bhate and Subhash Kak published an article on "Pāṇini's grammar and computer science". They defined a Pāṇinian rule as follows:

> A *Ps* [Pāṇini *sūtra*] is a single clause proposition consisting of a subject, a predicate, and an environment. It is a statement about grammatical features such as a suffix, an augment, a substitute, accent, reduplication, elision, and compounding. It is usually of the form *A* is *B* in the environment *C*. This can be written in the following formula:
>
> $$Ps\text{:}\ A \longrightarrow B\ (C)$$
>
> Here $\longrightarrow$ stands for *is* or *becomes*, and ( ) stands for *when*, *A* stands for the subject, *B* represents predicate, and *C* stands for environment. While *A* and *B* are the necessary components of a *sūtra*, *C* is optional (Bhate and Kak 1993 p. 5).

According to the authors, the three categories *A*, *B* and *C* can be either a single member or multiple member categories or a combination of both. An example of one member category which they provide is the rule: *iko yaṇaci*[17]. It can be represented by

$$A \longrightarrow B\,(C)$$

A multi-member category for *A* is the rule: *karmaṇyaṇ*[18]. This is noted as:

$$A^{1-n} \longrightarrow B\,(C)$$

---

[15] For a history of development of this model and critical review, see (Houben 1999 p. 41-46).

[16] Note the following remarks: "Since it reproduces standard speech, the A. [Aṣṭādhyāyī] is a prescriptive grammar. It states the rules which must be applied, if the speaker wants to convey meaning in a grammatically correct form. It is also a generative grammar, in two senses. First in this (Chomskyan) sense that in the process of derivation the wordform is fully described. Secondly, in the sense that, with the help of a limited number of rules (about 4000), and with the help of the *dhātupāṭha* and *gaṇapāṭha*, which provide the basic lexical elements, the A. is able to produce an infinite number of words, and thus, an infinite number of sentences." (Joshi and Roodbergen 1991 p. 15-16). Joshi uses the term generative grammar taken from the Chomskyan context, but at the same time he clarifies that it be used in a slightly different manner in Pāṇinian system (Joshi 1968 p. ix.fn.22). See also (Joshi and Roodbergen 1980 p. vi-xv).

[17] इको यणचि ॥६.१.७७॥ ▶ i*k* is replaced by ya*ṇ* if it is followed by a*c*.

[18] कर्मण्यण् ॥३.२.१॥ ▶ if a pada functioning as karman occurs together with a dhātu then suffix a(ṇ) is introduced after the dhātu.

The nature of this formulation by Bhate and Kak is close to the context-sensitive rules mentioned earlier. The main contention of this representation, however, is a general one. It says that in the grammatical process a given element *A* attains a particular identity or is transformed to some other form *B*. This happens when there is some suitable condition (*C*). It does not take into account the details of the derivational process nor does it provide a practical framework to apply the rules. Moreover, it does not account for instances where the derivational history or earlier stages provide the conditions for some operation. Nor does it specify the different kinds of operations that are needed for the process of synthesis. Apart from a few examples to show the formal nature of some of the rules of Aṣṭādhyāyī, it fails to develop a workable model of the Pāṇinian processes.

In his article on the context-sensitive nature of Pāṇinian rules Staal clearly notes that this is only the case with a limited number of rules. To quote him:

> In the following we shall be concerned with some rules of Sanskrit grammar as described by Pāṇini, which are context-sensitive. It is neither suggested that such rules suffice for the description of Sanskrit grammar, nor that Pāṇini thought so (Staal 1965 p. 63-64).

Despite the cautious note of Staal, Bhate and Kak suggest the proximity of computer programs and Pāṇinian grammar, primarily on the basis of such rules. The following quotation by them is illustrative of this hypothesis:

> The rules [of Aṣṭādhyāyī] are of different kinds: some are universal and context-sensitive transformations, others operate sequentially or recursively. Generally these rules are expressed in three groups: (i) rules of interpretation or meta-rules-*sañjñā* and *paribhāṣā* rules, (ii) rules of affixation-rules prescribing affixes after two kinds of basic *dhātu* and *prātipadika* roots, and (iii) rules of transformation for the stems and the suffixes-the morpho-phonemic rules. Note that a computer program has exactly the same general features of context-sensitive rules, recursion, and sequential rule application. It is not surprising, therefore, that these *sūtra*s have been compared to a computer program that generates Sanskrit sentences. Pāṇini's grammar is algebraic where a finite set of rules generates an infinite number of words and sentences (Bhate and Kak 1993 p. 2).

They do not show the recursive nature of Pāṇinian rules, and the context-sensitive character, mentioned by them above, is not what Staal demonstrates for some phonemic substitutions only.[19]

Taking the above clue, Sridhar Subbanna and Shrinivasa Varakhedi, in their paper on the computational structure of the Aṣṭādhyāyī, mention that "[T]he structure [of Aṣṭādhyāyī] consists of definitions, rules, and meta-rules that are context-sensitive and operate in sequence or recursively (Subbanna

[19] To substantiate their claims, they further point out the principles of numerical correspondence 1.3.10, the idea of ellipsis (*anuvṛtti*), code-letters (*anubandha*) and the law of general and exceptional rules (*utsarga* and *apavāda*).

and Varakhedi 2009 p. 56)".[20] Following the same note, Pawan Goyal, Amba Kulkarni and Laxmidhar Behera posit the context-sensitive nature of *vidhi* rules. To quote them: "It has been already recognized that Pāṇini expresses all such rules as context sensitive rules (Goyal, Kulkarni and Behera 2009 p. 144,153)".[21]

The claims of Peter Scharf and Malcolm D. Hyman about the XML and Pearl scripts they wrote for *sandhi*, nominal and verbal inflections, are modest in comparison to the above examples (Scharf 2009 p. 117-125). The authors note that "[W]e look forward to utilizing the enriched framework in a revised, faithful model of Pāṇinian declension. We are currently enriching the XML tagset further to allow derivation of participle stems and hope to go on to implement derivational morphology generally (Scharf 2009 p. 125)." Hyman introduces an XML vocabulary for expressing Pāṇini's *sandhi* rules (Hyman 2009 p. 253-265). XML, however, is again a framework to implement context-free grammars, which sometimes in their later variations, like XML-Schema, can be extended to represent context-sensitive rules (DeRose 1997 p. 105-106,139-142). The framework and the corresponding data structures, therefore, fall short of the potential to implement the rules where the conditions are formulated in a more complex manner than the immediate left or right contexts.

To conclude, scholars like Staal took inspiration from the generative grammar approach of Chomsky and tried to show that some of the rules of the Aṣṭādhyāyī correspond to the Chomsky hierarchy. The fact that grammars listed in the Chomsky hierarchy are suitable for computer languages, prompted some to hypothesize that the entire grammatical process can be written like a computer program. The above review shows that the recent attempts to computerize the Aṣṭādhyāyī emphasized the context-sensitive nature of Pāṇinian rules. There is, however, no study which establishes it to be sufficient for implementation of the whole of the Aṣṭādhyāyī on computer. Pāṇini's work with a formal structure that "can be easily adapted so as to perform numerical processing" (Bhate and Kak 1993 p. 2) is still waiting for computer implementation. In this regard, the following remark of J. E. M. Houben made some years ago, is still pertinent:

> Since at least twenty years there have been ideas to develop "programs replicating Pāṇinian prakriyā" and programs that analyse "strings in terms of Pāṇinian rules" (cp. Cardona 1999 : 272f). Inspite of several elaborate and sophisticated attempts in this direction, it seems we are still far from a comprehensive and convincing endresult. Why is it proving so difficult, for at least some twenty years, to computerize

[20] Kiparsky (2002) seems to be misquoted here by Subbanna and Varakhedi. The statement is from Bhate and Kak (1993 p. 2).

[21] By "all such rules" is meant here rules for "assigning a name, substitution, insertion, deletion". They refer to the previous study of Bhate and Kak (1993).

> Pāṇini's grammar? Perhaps a major reason is that we are not clear on some crucial issues regarding Pāṇini's grammar (Houben 2009 p. 18).

The above scepticism is shared by other experts of Pāṇinian grammar. During his keynote address at the "Third International Symposium on Sanskrit Computational Linguistics" at Hyderabad in 2009, S. D. Joshi made the following remark:

> Reading statements about information coding in which Pāṇini is hailed as an early language code information scientist, I am reminded of the situation in the early sixties, after Chomsky had published his book on Syntactic Structures in 1957. Here Chomsky introduced a type of grammar called transformational generative grammar. It earned him a great of applause, globally, I may say. Then it dawned on linguists that Pāṇini had also composed a generative grammar. So Pāṇini was hailed as the fore-runner of generative grammar. That earned him a lot of interest among linguists. Many linguists, foreign as well as Indian, joined the bandwagon, and posed as experts in Pāṇinian grammar on Chomskyan terms. Somewhat later, after Chomsky had drastically revised his ideas, and after the enthusiasm for Chomsky had subsided, it became clear that the idea of transformation is alien to Pāṇini, and that the Aṣṭādhyāyī is not a generative grammar in the Chomskyan sense. Now a new type of linguistics has come up, called Sanskrit Computational Linguistics with three capital letters. Although Chomsky is out, Pāṇini is still there, ready to be acclaimed as the fore-runner of Sanskrit Computational Linguistics (Joshi 2009 p. 1).

It should be noted here that some scholars have expressed their disagreement regarding the very possibility of computer automation of the Aṣṭādhyāyī. Thus Frits Staal conjenctured in the year 1966: "The third stage, that of automation, …may not even be effectively realizable" (Staal 1966 p. 209). Hartmut Scharfe, notes recently some four decades after Staal:

> We have to reject, I believe, the idea that Pāṇini's grammar is, as it were, a machine that produces correct Sanskrit words and sentences, if only we apply its rules in conformity with established meta-rules of application (Scharfe 2009 p. 85).

1.2 Nature and scope of the present work

It is in the context of the above mentioned scepticism regarding the formalization and computerization of Aṣṭādhyāyī that the present study assumes its relevance. It must be mentioned at the outset that this study does not intend to identify or establish the presence of features of modern linguistics and computational linguistics in the Pāṇinian system of Sanskrit grammar. Neither does it seek to show that Pāṇini anticipates several modern approaches or that his Aṣṭādhyāyī can be considered as the "first computer". The main aim of the present effort is to explore the possibilities of a formal representation of the content and processes of the Aṣṭādhyāyī and to enquire into the potential of its computer implementation.

The first question that arises is whether the text corpus of Aṣṭādhyāyī is formal enough to allow direct computer implementation? In other words, would it be possible to write a program whose input is the text corpus of the Aṣṭādhyāyī, and whose output is, a representation, which a computer can interpret and apply. I put forward this question as the first hypothesis and call it *the strong version of the formalization hypothesis*.

If the above hypothesis is true, then it would imply that what one needs to undertake is to follow the Aṣṭādhyāyī in its text and spirit and devise programs to implement it *in toto*. And since the Aṣṭādhyāyī is interpreted and applied with the help of other later texts—like Siddhāntakaumudī for applicational considerations or Paribhāṣenduśekhara for meta-rules etc.—these can be taken into account for the purpose of computerization. In other words, the task would be to accurately *simulate* the traditional manner of grammatical representations and applicational procedures. Most of the ongoing projects on computerization of the Aṣṭādhyāyī work along these lines. In approaching the task of computerization in this manner, they attest to a tacit confidence in the feasibility of this hypothesis.

Such an approach undertaken by several scholars is understandable. After all, there is a well established tradition of Pāṇini and the Pāṇinīyas spanning over two and a half millenia which is a glaring testimony to its efficacy. This further substantiates the opinion that there is absolutely no scope for any kind of tampering with the established way of reading and applying it. This is also in accordance with the principle set by Patañjali in his Mahābhāṣya where he cautions against any such attempt.[22]

After examining the grammatical corpus my conclusion is that it would not be possible to write a computer program that can directly process the present corpus of the Aṣṭādhyāyī. This, however, does not imply that the Aṣṭādhyāyī completely lacks formal components. Pāṇini's work is an exemplary attempt to formulate the grammar in a formal manner. Research in the last few decades has adequately established this fact.[23] The point here is not whether or not the Aṣṭādhyāyī has formal components, but whether these are sufficient for a *direct* implementation on computer.

There are several challenges which I discuss in section 3.2. The main argument against such an approach is that it would entail considering the Aṣṭādhyāyī as a closed, complete and perfect device. This, however, is not the case. For example, we do not have any precise information regarding the rule boundaries in several cases. Moreover, there are later additions and

[22] See e.g. (PB. 158).

[23] See here specially the following publications: (Staal 1965, 1965a, 1975), (Petersen 2008) and (Kiparsky 2009).

emendations suggested e.g. by Kātyāyana. A closer look at the grammatical corpus suggests that it is judicious to consider it as an open, flexible and growing network of grammatical content and processes, based on some fundamental systematic methodology, which can accommodate additional information if it is needed for precise specification.

Another aspect that argues against the above hypothesis, is that the corpus of Aṣṭādhyāyī is composed with organizational optimality (brevity or *lāghava*) as its main goal. The application of grammar for a particular process is not explicitly mentioned and is left largely to the person using it. This, however, needs to be specified in an explicit manner in case computer implementation of the derivational process is desired.

At this point it becomes important to clarify the nature of the main task of this research. Formalization and computerization involve representing the content and processes of the Aṣṭādhyāyī in a new medium. The formulation of the corpus of Aṣṭādhyāyī is in Sanskrit with special meta-linguistic conventions. Moreover, it is meant for application by individuals who, after understanding and remembering its techniques, apply it for derivational procedures. Both these aspects—i.e. the oral framework and application by human individuals—change in case of a formal representation and computer implementation. Thus, while the content and processes remain the same, the manner and the medium in which these are comprehended or represented is different and the applicational agency is now not the learned human students of grammar, but computers or logical systems.

Acknowledging that the strong version of formalization hypothesis is not feasible and the current formulation of the Aṣṭādhyāyī was meant for oral transmission and application by human scholars, the next task is to explore the other options. One of the main aims of this study is to enquire into the questions: Does Aṣṭādhyāyī function in an algorithmic manner? If yes, then what is the nature of this algorithm? What approach may one take to make the algorithmic character of the Aṣṭādhyāyī explicit?

The first reaction—prevalent among several modern scholars of Aṣṭādhyāyī as well—is that it is an example *par excellence* of a perfect algorithm for generating standard Sanskrit expressions. The reason provided by them is that Aṣṭādhyāyī employs a highly developed meta-language that clearly specifies the rules of grammar and that linguistic expressions can be generated by applying these rules. Moreover, there is a well established tradition of grammarians—the Pāṇinīyas—contributing to the understanding of the Aṣṭādhyāyī.

Following this viewpoint, one may assume that Aṣṭādhyāyī consists of fixed structures that are represented in a consistent manner. These structures comprehend and explain the Sanskrit language. Further, the nature of these structures is algorithmic and is guided by the rules and meta-rules of Aṣṭādhyāyī. Any Sanskrit expression can be derived by following an algorithmic procedure. It would involve applying the relevant rules one after another. One may implement this task through a computer program which would involve telling the computer what to do next. The program would judge whether a particular rule is applicable or not and execute it accordingly. The task here is primarily of a technical nature.

The second stage is that such a program could—to a large extent—interpret and decide, what is to be done next. This is a more demanding task. The underlying assumption is that there is some *principled system* that guides the dynamics of the derivational process, some inherent order based upon which derivational stages can be interpreted and right decisions can be made. At this point it becomes important to ask about the manner in which the grammatical corpus is articulated and the way in which the derivational process is executed. How far the tasks which are formulated in a specialized Sanskrit and are designed and meant for human application can be transferred to the computers? In which manner? And what is the way to do it?

Several issues are involved here. Whether the manner in which the grammatical system is articulated in Aṣṭādhyāyī is feasible and suitable for computers or not. Whether the manual application of a grammatical system for the purpose of deriving linguistic expressions can be simulated or not. How much and in which manner the task of decision-making can be invested in a non-human logical apparatus? In other words, what kind of model for the Pāṇinian system is most suitable for the purpose of its computer implementation?

Given the opposing views between some experts of the Pāṇinian system and the researchers attempting to automate the Aṣṭādhyāyī (see section 1.1), I intend to approach the task of formalization and computer implementation differently. Instead of attempting to automate the Aṣṭādhyāyī directly, I suggest first looking into the underlying systematic approach on the basis of which grammar is constructed. The systematic approach is to be gleaned by examining the descriptive methodology of ancillary disciplines.[24] Apart from the fact that they are associated with the Vedas, there is an evident commonality of approach with respect to their goals and methods. The prime effort of the ancillary disciplines is retention of a given phenomenon. For the Śrautasūtras this phenomenon may be a given sequence of ritual actions. The Prātiśākhyas, as well as *Śikṣā* and *Chandas* texts, aim towards retention of a

[24] The ancillary disciplines or the Vedāṅgas (lit. limbs of the Vedas) are: *Śikṣā* (phonetics), *Chandas* (prosody), *Vyākaraṇa* (grammar), *Nirukta* (etymology), *Kalpa* (instructions on ritual practice) and *Jyotiṣa* (astronomy). See: (Gonda 1975 p. 34).

specific collection of Vedic recitations. The Śulbasūtras provide retention of the plans and designs of the ritual arena etc. Similarly, the Aṣṭādhyāyī seeks to retain the standard usage of Sanskrit expressions.

For the purpose of retention of some given phenomenon, the ancillary disciplines follow a systematic method, which again is common to all of them. In order to substantiate the proposition that there does exist such an underlying general system of description of a given phenomenon, which permeates across the ancillary disciplines, I have worked-out a few detailed examples in appendix A.1.

Seen from the systematic point of view, grammar also follows the same goals and methods. There is no fundamental difference between Pāṇini's system and the systems developed in other ancillary disciplines. This also explains why technical terms from other disciplines could be easily borrowed and utilized in the grammar.[25] Moreover, amendments and extensions in the grammatical corpus indicate the presence of an underlying system which facilitates flexibility and portability.

Although apparently not so spectacular, the proposition that there is a common underlying system across the ancillary disciplines has significant consequences for formalization of the Aṣṭādhyāyī. The grammatical corpus can now be considered as a presentation of this general system in a particular framework. It is the framework which specifies how components are enunciated in the corpus, how are they characterized and combined. Further, how the entire mechanism is organized and applied, as well as how it is communicated is also dependent on the framework. Thus, I propose to distinguish between the general system of grammar and the framework in which it is presented. The corpus of Aṣṭādhyāyī, one can now assert, is the general system presented in a special framework. One may call this special framework the *Pāṇinian framework*.

The strong version of formalization hypothesis can now be reformulated as follows: the Pāṇinian framework is not sufficiently adequate for formalization. It does not negate or ignore the fact that the Pāṇinian framework is a wonderful example of a major effort to present the general system in a formal manner. Yet it is not *adequate* for a computer implementation.[26]

---

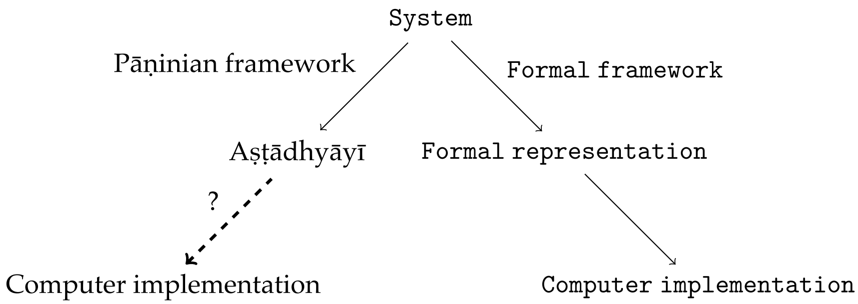
[25] Examples include *anusvāra*, *ṣaṣṭhī*, *saptamī* etc. There are some 50 terms which Pāṇini uses without defining them. For a list, see (Subrahmanyam 1999 p. 109-163).

[26] One may select a consistent and adequate core that can be formalized, akin to S. D. Joshi's proposal to consider the systematic consistent portions to be the original core and the incompatible parts as later additions. See: (Joshi and Bhate 1984 p. 252-253). If one is adamant to sift out parts of Aṣṭādhyāyī that would conform to a formal representation, then it is another matter. In that case, however, to identify these portions, one would eventually require such a formal framework!

The task which lies ahead is to evolve a new formal framework in which the contents and processes of the Aṣṭādhyāyī can be represented.

Here, I put forward and intend to establish, another *weak version of the formalization hypothesis*. Its main propositions can be stated as follows: the grammatical system followed by the Aṣṭādhyāyī is an outcome of a common systematic approach followed by the ancillary disciplines (Vedāṅga) associated with the Vedic corpus. The functioning of the general grammatical system can be represented in a formal manner. For this a new formal framework would be required. Pāṇinian content and processes can then be *re-presented* in this framework. The new framework, being formal in nature, can also be implemented on computer.

The tasks mentioned above are organized in the following manner.



At the top of the diagram is the general system which I outline in chapter 2. This, I propose, constitutes the core methodology of the grammatical processes. It is not grammar but the basic methodological system on which the grammar is specified.

The Aṣṭādhyāyī of Pāṇini can now be seen as a formulation of this system in a particular framework, which I call the Pāṇinian framework. The nature and characteristics of this framework are known to us from extensive scholarly research on the Aṣṭādhyāyī and is not the main focus of the present study. In sec. 3.2 some of the problems are mentioned which one would face if one were to attempt a direct implementation of the corpus of Aṣṭādhyāyī. In the above diagram it is noted by a dashed arrow with a question mark.

I, however, propose to undertake a *re-presentation* of the grammatical system in a different framework, which I call a formal framework. This new framework is introduced in chapter 3. Chapter 4 provides a formal representation of Aṣṭādhyāyī in terms of statements that are formulated in the new framework. Finally, chapter 5 provides the algorithms for a possible computer implementation of the statements of the Pāṇinian system of Sanskrit grammar.