

+a:—
Xm+a || Xm[—]
+a:—
X+a || X[—]
+a: guṇa
+a: vṛddhi
1. Xm+a || Xm[at, eṅ]
+a: vṛddhi
1. Xm+a || Xm[āt, aic] 1. अदेशुणः || १.१.२ || āt, aic
+a:—
वृद्धिरादैच् || १.१.१ || +a:—
1. X+a || X[—] 1. X<y || X[—] ^ yM[—]
+a: guru(p) Xm+a || Xm[—] X<y || X[—] ^ yM[—]
1. Xm+a || Xm[hrasva] ^ Xn[samyoga] +a: vṛddhi
तरसप्तमपौ घः || १.१.२२ || 1. Xm+a || Xm
<y: l(a)(ṭ) 2. Xm+a || Xm[dīrgha] +a: gha 1. Xm+a || Xm
1. X<y || X[dhātu] ^ yM[vartamāna] 1. X+a || X[tara(p), tama(p)]
+a: laghu
वर्तमाने लट् || ३.२.१२३ || वृद्धिरादैच् तरसप्तमपौ घः || १.१.२२ ||
1. Xm+a || Xm[—] ^ XnNO: +a:—
+a:— X<y || X[—] ^ yM[—]
ह्रस्वं लघु || १.४.१०० || संयोगं गुरु || १.४.११ Xm+a || Xm[—] ^ XnNOT[—]
X<y || X[—] ^ yM[—] 1. XmXn+a || Xm[ha] ^ Xn[ha] ^ IDX[n=m+1]
+a:— +a:—
+a:— Xm+a || Xm[—]
ह्रोलोऽनन्तराः संयोगः || १.१.७ || Xm+a || Xm[—]
Xm+a || Xm[—] ^ Xn[—]
Xm+a || Xm[—]

Modeling the Pāṇinian System of Sanskrit Grammar

Anand Mishra

HEIDELBERG
UNIVERSITY PUBLISHING

Modeling the Pāṇinian System
of Sanskrit Grammar

Anand Mishra

Modeling
the Pāṇinian System
of Sanskrit Grammar

HEIDELBERG
UNIVERSITY PUBLISHING

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at <http://dnb.dnb.de>.



This book is published under the Creative Commons Attribution 4.0 License (CC BY-SA 4.0). The cover is subject to the Creative Commons License CC BY-ND 4.0.

The electronic open access version of this work is permanently available on Heidelberg University Publishing's website: <https://heiup.uni-heidelberg.de>
urn: urn:nbn:de:bsz:16-heiup-book-294-9
doi: <https://doi.org/10.17885/heiup.294.395>

Cover illustrations: © Anand Mishra.

Text © Anand Mishra, 2019.

ISBN 978-3-946054-55-9 (Softcover)
ISBN 978-3-946054-54-2 (Hardcover)
ISBN 978-3-946054-53-5 (PDF)

अये लसत्पाणिनिचारुचन्द्रक्षरत्सुधापानचकोरचित्ताः ।
बुधा भवत्कर्णनिकेतनेषु मद्दुक्तवाक्यान्यतिथीभवन्तु ॥

For my father
Prof. Dr. Kedar Nath Mishra
(1935–1999)

Contents

1	Introduction	7
1.1	Earlier research	7
1.2	Nature and scope of the present work	15
2	System	21
2.1	The grammatical cycle: analysis and synthesis	22
2.1.1	Why grammar?	25
2.2	Grammatical synthesis	32
2.2.1	Form and content of a component	32
2.3	The derivational process	44
2.3.1	Guiding principles of synthesis	44
2.3.2	Intention of the speaker	45
2.3.3	Consistency of the derivational state	45
2.3.4	Saturation of the derivational state	46
2.3.5	Completion of the derivational state	47
2.3.6	Conditions	48
3	Frameworks	49
3.1	Salient features of the Pāṇinian formulation	50
3.1.1	Methodology of Pāṇinian formulation	52
3.2	Formalization of the Pāṇinian corpus: challenges and possibilities	57
3.2.1	Possibility of formalization of the grammatical system ..	62
3.3	Specification of the formal framework	66
3.3.1	Representing a component	68
3.3.2	Comprehending the dynamics of the derivational process	74

4	Representations	79
4.1	Operational statements	79
4.1.1	Characterizing the components	79
4.1.2	Combining the components	95
4.2	Structure of a statement	98
4.2.1	Types of operations	99
4.2.2	Complexity of a statement	100
4.3	Relation between statements	104
5	Implementation	107
5.1	Data-structures	108
5.1.1	Elements	108
5.1.2	Sound-sets	109
5.1.3	Language-components	111
5.1.4	Sentences	113
5.1.5	Derivational states	113
5.1.6	Slices	114
5.1.7	Process-strips	114
5.1.8	Statements	115
5.1.9	Statement groups	116
5.2	Processes of grammar	117
5.2.1	Initialisation	118
5.2.2	Saturation	120
5.2.3	Completion	124
A	The Systematic Approach	131
A.1	Analysis and synthesis	132
A.1.1	Syllables (<i>Akṣaras</i>)	132
A.1.2	The falcon shaped fire altar (<i>Śyenacit</i>)	134
A.1.3	Samhitā-pāṭha and Pada-pāṭha	136
A.2	Components and units	139
A.2.1	Characterization	141
A.2.2	Combination of units and components	142
A.2.3	Constructs	142
A.3	Structures, variations and change	143

Abbreviations and Conventions

- BU Bṛhadāraṇyakopaniṣad
BŚuS Baudhāyanaśulbasūtra
KV Kāśikāvṛtti
KŚS Kātyāyanaśrautasūtra
MB Mahābhāṣya
MU Muṇḍakopaniṣad
N Nirukta
PŚ Paribhāṣenduśekhara
P Paspasāhnikā
PB Paspasāhnikabhāṣya
PV Paspasāhnikavārtika
ṚV Ṛgveda
ṚVPr Ṛgvedaprātiśākhya
ŚB Śatapathabrāhmaṇa
TU Taittirīyopaniṣad
VS Vājasaneyisaṃhitā
VPr Vājasaneyiprātiśākhya
V Vārttika
YS Yogasūtra
YSB Yogasūtrabhāṣya

Conventions

Numbers like 1.1.26 refer to the *sūtras* of Aṣṭādhyāyī. Sanskrit words and sentences in the main body of the book are in *italics*. Sanskrit sentences and the *sūtra*-texts in footnote are in देवनागरी. Components and other elements of grammar are in normal font. Terms like sound-sets that are introduced by me for the description of the formal framework are in typewriter font. Indicatory “it-markers” appear in brackets, e.g. ti(p). Phonemes for the sake of pronunciation are in square brackets, e.g. n[u](m).

Preface

The present work is a study of the Aṣṭādhyāyī of Pāṇini from a new perspective and is an adapted version of my doctoral dissertation with the same title. It attempts to explore the Pāṇinian grammar from a formal point of view and investigate the possibilities of representing it in a logical, explicit and consistent manner. Such a representation requires an appropriate framework. A formal framework would facilitate adequate tools for postulating and evaluating hypotheses about the grammatical system. Moreover, it would furnish the basis for a computer implementation of the grammar. Both aspects, namely a formal representation and computer implementation of the Aṣṭādhyāyī, are objects of enquiry in the field of theoretical studies on Pāṇini as well as the emerging discipline of Sanskrit computational linguistics. This book takes on the ground-work in these areas.

The propositions that I put forward in this book are a result of my experimentations with the Aṣṭādhyāyī. Over the last few years, I tried a number of models to comprehend the content and processes of the Pāṇinian system. Beginning with the initial aim to automatize Aṣṭādhyāyī, I examined the various challenges and issues accompanying this and in the process graduated to work on the development of a formal framework for the grammar. The outcome of this ongoing process is summarized in the present work.

There are several excellent expositions of Aṣṭādhyāyī by the scholars in this field and these are evidently the chief source of my understanding on this subject. At the same time, there is hardly any significant work on Pāṇini from a formal perspective. The relatively limited writings available are largely in the nature of unproven hypotheses with few exemplary comparisons and usually with claims to show that Pāṇini has been the forerunner in matters of logic, mathematics and computer-science. The present work does not strive for any of the above objectives. It is not an attempt to compare the Pāṇinian system with modern grammatical theories. Neither is it an attempt

to establish Pāṇini as the source of the concepts and methods followed by modern computing systems. Neither does it claim that Sanskrit is the most suitable language for computers.

The present study attempts to render the Pāṇinian system of Sanskrit grammar in a framework which consists of unambiguous, consistent and explicit categories. Only then can it be conveyed to logical systems like modern day computers. Differing from the formulation of Aṣṭādhyāyī, which is composed in an artificial yet natural language and is meant to be employed by individuals who are acquainted both with the Sanskrit language and the techniques of grammar, the present rendering aims for a non verbal representation in terms of mathematical categories and logical relations which can be implemented in an algorithmic manner.

The process of formalization, however, involves determination of the underlying principles regarding the functioning of grammar. My first response was to look into the explanatory literature associated with the Aṣṭādhyāyī and on this basis to decide upon the general principles that may lead towards formalization. Although a careful study of the literature is of immense importance to enable understanding of the various issues, in my case it did not suffice to devise a solution for formalization, based only upon comparative studies of Pāṇinian literature. The reason lies primarily in the different nature of the task at hand. Later literature on Aṣṭādhyāyī is primarily explanatory.¹ Although there are significant attempts to clarify, uphold and sometimes rectify the grammatical corpus, there is hardly any effort to render it in a new formal setup. This is evident because the Aṣṭādhyāyī is (to a significant extent) a formal presentation of the grammar. And a very brilliant one indeed.²

The remarkable success of Aṣṭādhyāyī had the consequence that the main effort of later grammarians was directed towards keeping it intact. Apart from the attempts by grammarians like Rāmacandra (late 14th–15th century) and Bhaṭṭoji Dīkṣita (late 16th–17th century) to reorganize the rules or *sūtras* of Aṣṭādhyāyī with process or application (*prakriyā*) as the main focus, there are hardly any works dedicated to recasting Aṣṭādhyāyī.³

¹ On the explanatory nature of the literature on Aṣṭādhyāyī, see (Bhattacharya 1955 p. 123-132) and for a bibliographical note (Cardona 1980 p. 278-293).

² Patañjali considers Pāṇini to be an “embodiment of authority” प्रमाणभूत आचार्यः। (MB on 1.1.1) regards “the *sūtras* of Pāṇini as beautiful” शोभना खलु पाणिनेः सूत्रस्य कृतिः। (MB on 2.3.66) and postulates that “just because of doubtful appearance of a rule, it should not be rejected, but should be made precise on the basis of reasoned explanation” व्याख्यानतो विशेषप्रतिपत्तिर्नहि संदेहादलक्षणम्। (PB. 68). The Pāṇinīyas or the scholarly followers of Pāṇini formulate a number of conventions to keep intact the systematic coherence of Aṣṭādhyāyī, see (Wujastyk 1993).

³ Bhaṭṭoji Dīkṣita’s Siddhāntakaumudī is a reorganization of the Pāṇinian rules. Compared to the earlier attempt of Rāmacandra in his Prakriyākaumudī, Bhaṭṭoji Dīkṣita sticks to the

The question of formalization, in terms of a logical language which a computer program without any knowledge of Sanskrit could understand, was naturally not a requirement at that time either. Formal representational techniques that are being evolved today and increasingly being employed to develop machines and computers were missing, and it would be anachronistic and wrong to expect Pāṇini (or anyone) to anticipate the requirements and expectations of a computer program trying to implement it two-and-a-half millenia later. It should be noted here that I am neither denying the formal nature of Aṣṭādhyāyī, nor examples of precursors of several modern techniques in it.⁴ However, what is certain is that Aṣṭādhyāyī, as we have it today, would require considerable additional information organized in a suitable manner in order to make it executable as a computer program.⁵

As a consequence, a new methodological approach becomes necessary—the scientific method of observation, hypothesis and testing in order to develop a new formal representation of the Aṣṭādhyāyī.⁶ The present work is primarily based on this methodological approach. Unlike most of the publications on Pāṇinian grammar, I do not attempt to comment upon an issue by collecting and analyzing the views of the traditional scholars or the Pāṇinīyas. Although an important task, this is hardly fruitful in my case. The reason for this is that I am confronted with a challenge of our time and the Pāṇinian tradition, because of its antiquity, had no occasion to comment on it. On the other hand, my investigations are directly based on the corpus of Aṣṭādhyāyī. My use of later Pāṇinīyas is in order to understand the tradition and to relate them to important premises of my hypotheses.

It could be asked here: how justified I am in proposing systematic frameworks that are not directly supported and employed by the Pāṇinīyas? I feel that Aṣṭādhyāyī is a text which not only allows but invites such an approach. Its composition is the result of an empirical observation and systematic organization of the linguistic features. It is an appropriate case for studying the methods of comprehending linguistic phenomena through developing systematic structures.

Another ground for formalization is that it has both theoretical and practical relevance in the field of Pāṇinian studies. Practically speaking, it facilitates better access to the content and processes of the Pāṇinian system, not just to experts in this field, but to non-Sanskritists as well. Theoretically speaking, it

Pāṇinian rules only. He also covers all the rules of the Aṣṭādhyāyī. See (Cardona 1980 p. 285-288) for bibliographical notes and (Houben 2008 p. 563-574) for the reasons for the tremendous success of Bhaṭṭoji Dīkṣita.

⁴ See section 1.1 for recent works on the formal character of Pāṇinian rules.

⁵ I will discuss this in section 3.2.

⁶ On the scientific approach, see (Wilson 1952) and (Popper 1959).

prompts us to look at the Aṣṭādhyāyī from a formal perspective. This entails a critical examination of the content and processes of the grammatical system. Strategies for evolving such a representation involve a reworking of the oral framework in which Aṣṭādhyāyī is composed. This leads to identification, analysis and determination of issues related to organization and application of Pāṇinian rules. A formal representation also facilitates precise formulation and testing of hypotheses regarding some of the fundamental issues of the Pāṇinian system—meta-linguistic conventions (*paribhāṣās*), brevity (*lāghava*), rule organization (*adhikāra* and *anuvṛtti*), ordering and application of grammar (*prakriyā*), functioning of the system etc. In short, it opens up a new paradigm for Pāṇinian studies.

to establish Pāṇini as the source of the concepts and methods followed by modern computing systems. Neither does it claim that Sanskrit is the most suitable language for computers.

The present study attempts to render the Pāṇinian system of Sanskrit grammar in a framework which consists of unambiguous, consistent and explicit categories. Only then can it be conveyed to logical systems like modern day computers. Differing from the formulation of Aṣṭādhyāyī, which is composed in an artificial yet natural language and is meant to be employed by individuals who are acquainted both with the Sanskrit language and the techniques of grammar, the present rendering aims for a non verbal representation in terms of mathematical categories and logical relations which can be implemented in an algorithmic manner.

The process of formalization, however, involves determination of the underlying principles regarding the functioning of grammar. My first response was to look into the explanatory literature associated with the Aṣṭādhyāyī and on this basis to decide upon the general principles that may lead towards formalization. Although a careful study of the literature is of immense importance to enable understanding of the various issues, in my case it did not suffice to devise a solution for formalization, based only upon comparative studies of Pāṇinian literature. The reason lies primarily in the different nature of the task at hand. Later literature on Aṣṭādhyāyī is primarily explanatory.¹ Although there are significant attempts to clarify, uphold and sometimes rectify the grammatical corpus, there is hardly any effort to render it in a new formal setup. This is evident because the Aṣṭādhyāyī is (to a significant extent) a formal presentation of the grammar. And a very brilliant one indeed.²

The remarkable success of Aṣṭādhyāyī had the consequence that the main effort of later grammarians was directed towards keeping it intact. Apart from the attempts by grammarians like Rāmacandra (late 14th–15th century) and Bhaṭṭoji Dīkṣita (late 16th–17th century) to reorganize the rules or *sūtras* of Aṣṭādhyāyī with process or application (*prakriyā*) as the main focus, there are hardly any works dedicated to recasting Aṣṭādhyāyī.³

¹ On the explanatory nature of the literature on Aṣṭādhyāyī, see (Bhattacharya 1955 p. 123-132) and for a bibliographical note (Cardona 1980 p. 278-293).

² Patañjali considers Pāṇini to be an “embodiment of authority” प्रमाणभूत आचार्यः। (MB on 1.1.1) regards “the *sūtras* of Pāṇini as beautiful” शोभना खलु पाणिनेः सूत्रस्य कृतिः। (MB on 2.3.66) and postulates that “just because of doubtful appearance of a rule, it should not be rejected, but should be made precise on the basis of reasoned explanation” व्याख्यानतो विशेषप्रतिपत्तिर्नहि संदेहादलक्षणम्। (PB. 68). The Pāṇinīyas or the scholarly followers of Pāṇini formulate a number of conventions to keep intact the systematic coherence of Aṣṭādhyāyī, see (Wujastyk 1993).

³ Bhaṭṭoji Dīkṣita’s Siddhāntakaumudī is a reorganization of the Pāṇinian rules. Compared to the earlier attempt of Rāmacandra in his Prakriyākaumudī, Bhaṭṭoji Dīkṣita sticks to the

The question of formalization, in terms of a logical language which a computer program without any knowledge of Sanskrit could understand, was naturally not a requirement at that time either. Formal representational techniques that are being evolved today and increasingly being employed to develop machines and computers were missing, and it would be anachronistic and wrong to expect Pāṇini (or anyone) to anticipate the requirements and expectations of a computer program trying to implement it two-and-a-half millenia later. It should be noted here that I am neither denying the formal nature of Aṣṭādhyāyī, nor examples of precursors of several modern techniques in it.⁴ However, what is certain is that Aṣṭādhyāyī, as we have it today, would require considerable additional information organized in a suitable manner in order to make it executable as a computer program.⁵

As a consequence, a new methodological approach becomes necessary—the scientific method of observation, hypothesis and testing in order to develop a new formal representation of the Aṣṭādhyāyī.⁶ The present work is primarily based on this methodological approach. Unlike most of the publications on Pāṇinian grammar, I do not attempt to comment upon an issue by collecting and analyzing the views of the traditional scholars or the Pāṇinīyas. Although an important task, this is hardly fruitful in my case. The reason for this is that I am confronted with a challenge of our time and the Pāṇinian tradition, because of its antiquity, had no occasion to comment on it. On the other hand, my investigations are directly based on the corpus of Aṣṭādhyāyī. My use of later Pāṇinīyas is in order to understand the tradition and to relate them to important premises of my hypotheses.

It could be asked here: how justified I am in proposing systematic frameworks that are not directly supported and employed by the Pāṇinīyas? I feel that Aṣṭādhyāyī is a text which not only allows but invites such an approach. Its composition is the result of an empirical observation and systematic organization of the linguistic features. It is an appropriate case for studying the methods of comprehending linguistic phenomena through developing systematic structures.

Another ground for formalization is that it has both theoretical and practical relevance in the field of Pāṇinian studies. Practically speaking, it facilitates better access to the content and processes of the Pāṇinian system, not just to experts in this field, but to non-Sanskritists as well. Theoretically speaking, it

Pāṇinian rules only. He also covers all the rules of the Aṣṭādhyāyī. See (Cardona 1980 p. 285-288) for bibliographical notes and (Houben 2008 p. 563-574) for the reasons for the tremendous success of Bhaṭṭoji Dīkṣita.

⁴ See section 1.1 for recent works on the formal character of Pāṇinian rules.

⁵ I will discuss this in section 3.2.

⁶ On the scientific approach, see (Wilson 1952) and (Popper 1959).

prompts us to look at the Aṣṭādhyāyī from a formal perspective. This entails a critical examination of the content and processes of the grammatical system. Strategies for evolving such a representation involve a reworking of the oral framework in which Aṣṭādhyāyī is composed. This leads to identification, analysis and determination of issues related to organization and application of Pāṇinian rules. A formal representation also facilitates precise formulation and testing of hypotheses regarding some of the fundamental issues of the Pāṇinian system—meta-linguistic conventions (*paribhāṣās*), brevity (*lāghava*), rule organization (*adhikāra* and *anuvṛtti*), ordering and application of grammar (*prakriyā*), functioning of the system etc. In short, it opens up a new paradigm for Pāṇinian studies.

Acknowledgements

My deep gratitude to my mother Sneh Lata Mishra and my father Prof. Dr. Kedar Nath Mishra for instilling in me the love for Sanskrit; to my teachers at the Indian Institute of Technology Kanpur for instructing me in logical thinking and mathematical formulations; to my supervisor Prof. Dr. Axel Michaels for making me think critically in an independent but responsible manner; to my friends and colleagues for their support and encouragement, and to the authors listed in the bibliography for their scholarly publications without which this work would not have been possible.

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 *kāraḥas* 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 *kāraḥa*-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.iit.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, *anuvṛttis*, *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 *sūtras*, 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: “*Pratyāhāras* or features? A qualitative analysis of phonological descriptive techniques—a comparison of Pāṇini’s *pratyāhāras* 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 Chomsky¹² 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 ॥ ► *ik* is replaced by *yaṇ* if it is followed by *ac*.

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: *yathāsamkhyamanudeśaḥ 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 succinct 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:

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

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. Roodbergen¹⁶—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:

$$Ps: A \longrightarrow B (C)$$

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: *karmanyaṇ*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 ॥ ► *ik* is replaced by *yaṇ* if it is followed by *ac*.

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 *sūtras* 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). In spite 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 conjectured 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 Prātiśākhya, 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

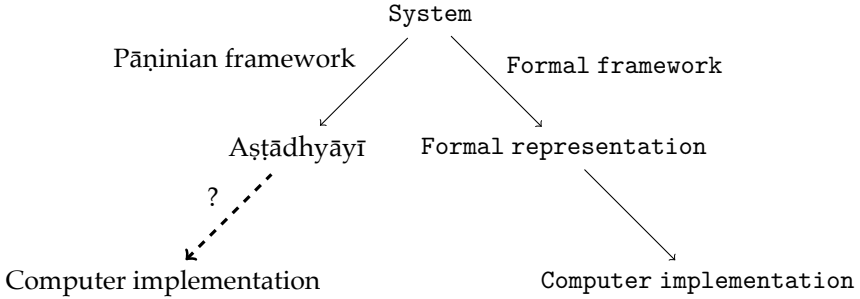
²⁵ Examples include *anusvāra*, *ṣaṣṭhi*, *saptamī* etc. There are some 50 terms which Pāṇini uses without defining them. For a list, see (Subrahmanyam 1999 p. 109-163).

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.

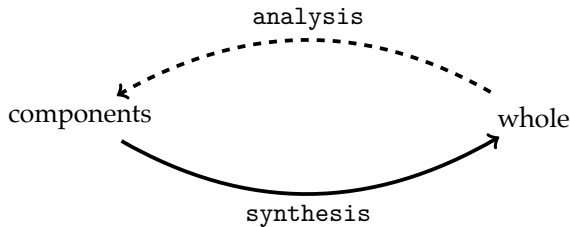
I, however, propose to undertake a *re-presentation* of the grammatical system in a different framework, which I call a formal framework. This new framework is introduced in chapter 3. Chapter 4 provides a formal representation of Aṣṭādhyāyī in terms of statements that are formulated in the new framework.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

Chapter 2

System

One of the prime goals pursued by the ancillary disciplines associated with the Vedas is retention of a given phenomenon.¹ By a given phenomenon, I mean any existing linguistic or cultural practice established over a number of generations. It is something which one has received as the standard and would like to protect it and pass it on to the next generation—for example, the recitation of the Vedic *mantras*, performance of rituals or linguistic usage.

In order to achieve the goal of retention, these disciplines also follow a common systematic approach which consists of two interdependent and complementary processes. The first one is *analysis* of a given whole into constituent components. The other one is *synthesis* through (rule based) combination of components to regain the given whole.



For example, the continuous recitation (*saṃhitā-pāṭha*) is analyzed into a word-for-word recitation (*pada-pāṭha*) and the Prātiśākhya texts provide a rule based synthesis from word-for-word recitation to the continuous recitation. Similarly, the Śulbasūtras provide the rules for preparing the ritual

¹ The six ancillary disciplines associated with the Vedas are said to be phonetics (*Śikṣā*), prosody (*Chandas*), etymology (*Nirukta*), grammar (*Vyākaraṇa*), instructions on ritual practices (*Kalpa*) and astronomy (*Jyotiṣa*) (Gonda 1975 p. 34). For a summary of the literature on phonetics and grammar, see (Scharfe 1977), on ritual practices (Gonda 1977) and for astronomy (Pingree 1981).

altar. The building blocks of the geometrical figures (squares, rectangles, triangles etc.) are gained through the process of analysis. Similarly, once the individual steps of a particular ritual are identified, these are arranged in the Śrautasūtras and one can re-constitute the ritual by following the rules mentioned there.

This apparently cyclical exercise comprehends a given phenomenon in a systematic manner and gives rise to an interconnected *structure* of components. Such structures have the tendency to last longer and are explained on the basis of the underlying *system*. Moreover, structures facilitate variations and change.

In section A.1 of the appendix, I have worked out a few examples to show the details of these processes of analysis and synthesis. In this chapter, we will focus on the special case of linguistic expressions and the grammatical cycle.

2.1 The grammatical cycle: analysis and synthesis

There is a consensus among Sanskrit grammarians, both ancient and modern, that grammar involves primarily an analysis of a given linguistic expression into its constituent components. Analysis follows the process of concurrent presence (*anvaya*) and concurrent absence (*vyatireka*).² The process of analysis is called *anvākhyanā* and it has two stages:

1. *Vākyavibhajyānvākhyanā* is analysis of sentences (*vākya*) and identifying its component words (*padas*). For example, analysis of the continuous recitation of a Vedic *mantra* or a Sanskrit sentence into constituent words.
2. *Padavibhajyānvākhyanā* is further analysis of individual *padas* in its constituents. This stage of analysis yields more granular components.³

For example, the following sentence having two *padas* :

² The term *vyākaraṇa* for grammar, whose etymological meaning is analysis, reflects these processes. It is used in the sense of separation of things in (ŚB. 1.7.1.4) and (KŚS. 4.2.7-9), in the sense of discrimination between something, for example, between *satya* and *anṛta* as in (VS. 19.77), as something which was previously one single entity, that was not previously differentiated, becoming or being made differentiated in (BU. 1.4.7), or it can also signify the act of dividing, e.g. in (ŚB. 3.3.1.13) and finally in the sense of making something clear in (YSB. 3.17) or Vyāsabhāṣya on (YS. 3.17). George Cardona (1999 p. 564-576) provides a detailed exposition on the etymology and uses of *vyākaraṇa* as well as other expressions for grammar.

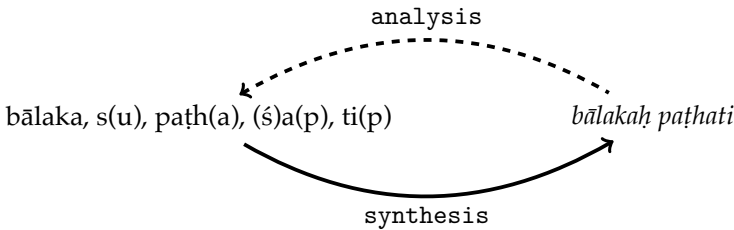
³ R. S. Bhattacharya (1966 p. 212-237) discusses these processes in detail. See also (S. D. Joshi 1968 p. xi).

bālakaḥ paṭhati (a boy recites)

is analyzed into constituent components such as:

bālaka, s(u), paṭh(a), (ś)a(p), ti(p).

The rules of grammar, however, establish a correlation between the analyzed components with the given linguistic expression. This is achieved by combining the components in a controlled manner to obtain the linguistic expressions. The grammatical system therefore not only undertakes an analysis of a given whole into components, but also consists of the complementary process of synthesis of the analyzed components to regain the original expression. This, in a way, is a cyclical exercise.



A given linguistic expression is first analyzed into constituent components and then re-gained by combining the components.

The process of synthesis is rule based. Rules lay down the constraints regarding the choices and manner in which the components should be combined. They prevent haphazard and unrestrained combinations and endorse the claim of the grammarians that the components *are* constituents of the original expression. The choice of components is vindicated by successful connection between them and the original expression.

In contrast to the rule-based nature of synthesis, there are no rules that bring about analysis of a given expression into constituent components.⁴ The components, however, are chosen in a manner so that they lead to the original expression once combined according to the rules of synthesis. The constituent components are stipulated by the grammarians for the purpose of grammar and exist within the grammatical system. They are not found in common usage and are, so to say, imaginary (*kālpānika*) i.e. improvised by the grammarians for their employment within the grammatical system.⁵

⁴ There are a few exceptions in the Prātiśākhya texts where such an attempt is made. See section A.1.3.

⁵ The grammatical components exist in the grammatical system (*śāstra*) only and not in the common usage (*loka*). Further, they are not established on the basis of usage (*lokataḥ siddha*) (R. S. Bhattacharya 1966 p. 213-214). Nāgeśa Bhaṭṭa in his *Vaiyākaraṇa-siddhānta-parama-laghu-mañjūṣā* remarks the imaginary nature of the components that are improvised by the teachers and employed only within the grammatical system (K.D. Shastri 1975 p. 7).

Moreover, the grammatical components are dependent on their parent expressions. This would be evident through an example Vyāsa provides in his commentary on *Yogasūtra* (3.17). The point he is noting there is that it is important to analyze the individual words further in order to determine, whether a word like *bhavati*, *aśvaḥ* or *ajāpayah* denotes an act or a *kāraka*.⁶ Here, *bhavati* can be either vocative singular feminine of the respectful pronoun *bhavatu*, and in this case consisting of the components *bhavatī-s(u)*,⁷ or third person singular of the present form with components *bhū-(ś)a(p)-ti(p)*. Similarly, *aśvaḥ* can have *aśva-s(u)* meaning “a horse”, or second person singular aorist form with *śvi-a(ñ)-si(p)* meaning “reached” and *ajāpayah* referring to an object (goat’s milk) has *ajāpayas-am* or the second person singular imperfect causative verb form of *ji* having *ajāp-i-a-s*. These examples show that the components are dependent on the parent expression or the given whole. If the given whole signifies an act, then the components are different from when it denotes a *kāraka*.

Not only are these components imaginary and dependent on their parent whole, they are also not unique. In other words, there may be more than one analysis of the same original expression depending upon the grammatical tradition. As an example, consider the expression *avati* (leads, brings to) which is derived using the root *av* according to the grammar of Whitney (1885 p. 4), but the root *u(ñ)* according to the Pāṇinian system. This implies that there can be more than one grammar of the same language.

The process of synthesis is provided through the rules of the *Aṣṭādhyāyī*. Without entering into the details of derivations, it suffices here to mention that the components are combined to create the given whole.⁸ Apart from that, the rules of grammar may also change depending upon the standard usage which they are supposed to account for.

Although the stipulated components are *non-real*—in that they come into existence through the grammarians and exist within the realm of grammar—this is not the case with the given linguistic expressions themselves. They are established (*siddha*) on the basis of their abiding usage among people.⁹

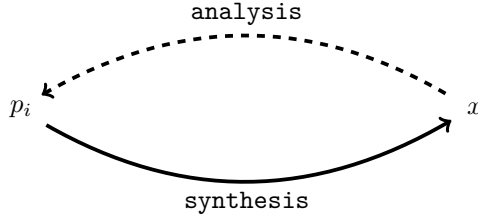
⁶ The word *kāraka* literally means “doer of an action” and in grammar used in a technical sense as “instrument of action” (Abhyankar 1986 p. 118). There are six *kārakas*. For details see (Cardona 1974 p. 231-306).

⁷ An example भवति भिक्षान्देहि comes in the commentaries to *Pāraskaragr̥hyasūtra* (2.5.2-4) and *Manusmṛti* (2.49).

⁸ S. D. Joshi (1968 p. ix-xi) notes this complementary character of the analysis (*vibhajya anvākhyāna*) and synthesis, combination or integration (*vṛtti*).

⁹ R. S. Bhattacarya (1966 p. 228-237) summarizes by noting that the constituent components of a Sanskrit expression are non-real (*kālpānika*), the rules of grammar (*upāya*) are not uniquely fixed (*anīyata*), but the linguistic expressions (*upeya*) are established (*siddha*).

Thus, the grammatical system as a whole consists of the complementary processes of analysis and synthesis. Any given standard linguistic expression (henceforth represented by the symbol x) is analyzed in constituent components (henceforth p_i). Aṣṭādhyāyī is a collection of the constituent components p_i together with the rules of synthesis. One can represent this cyclical process schematically as follows:



2.1.1 Why grammar?

A pertinent question arises here: Why should this cyclical procedure be undertaken at all? If grammatical processes involve regaining the original expression after one has analyzed it into components, then what is the use of such an exercise? In other words, what purposes are served by developing such a system?

The passages in the Paspasāhnika—both the Vārttika of Kātyāyana and the Bhāṣya by Patañjali—are explicit as to the purposes of grammar. Kātyāyana enumerates five:

- (i) *rakṣā* or safeguarding the Vedic texts as well as standard usage of the language (ii) *ūha* or suitable adaptation and modification of a *mantra* according to the requirements of a particular ritual (iii) *āgama* or complying with Vedic injunctions (iv) *laghu* or economy of effort and (v) *asaṁdeha* or absence of doubt as to the standard expressions are the purpose.¹⁰

The above enumeration of Kātyāyana and further explanations of Patañjali point out that the main purpose of developing the grammatical system is to safeguard the standard usage and effectuate acceptable modifications.

Both the non-Vedic common usage (*laukika*), as well as Vedic corpus, are sought to be preserved by the grammar. The Vedic corpus is a clearly delineated collection, and as to common usage, Pāṇinīyas recognize certain model speakers (the *śiṣṭas*) who set the standards. Patañjali in his Mahābhāṣya on 6.3.109 provides several characteristics of the model speakers on

¹⁰ रक्षोहागमलघ्वसन्देहाः प्रयोजनम्। (PV. 2).

the basis of their knowledge of the grammar as well as their dwelling place and behavior.

These model speakers are *brāhmaṇas* who dwell in the country of the *āryas* where alone their exemplary behavior is found. The *brāhmaṇas* who dwell in this abode of the *āryas* and have only as much grain as they can carry in a small pot, are not greedy for honor, following established rules of correct behavior without having to be given any immediate cause for this, and who have attained total expertise in some traditional area of learning without explicit instruction, these honorable ones are the *śiṣṭas*.¹¹

What is meant by correct speech or standard linguistic expressions, is the way these model speakers would speak.

The next question is how to safeguard standard usage. Patañjali's search for an answer to this question sheds important light on the nature of the grammatical method. The discussion on this point is in (PB. 46-55) and can be summarized as follows.¹²

One obvious way to retain the standard speech forms would be either to exhaustively specify them by uttering them one after another, or to stipulate the complementary set of non-standard forms. Both these options are to be discarded. It may be possible to retain a limited collection of utterances (as in case of the Vedic Saṃhitās) but impossible for the set of expressions of common speech as it is too large.¹³ Patañjali finally suggests providing a description of this big collection of standard expressions by pointing out their general and special characteristics and recording them in a systematic manner. His advice for attaining this is by putting constraints (*niyama*) that can distinguish between standard and non-standard speech.

In order to understand the concept of constraint or *niyama*, it is necessary to look into an important aspect regarding the nature of a linguistic expression, namely the distinction between its form and content. In Paśāhnikā (PB. 4-12) Patañjali looks into several suggestions regarding the exact nature of an expression like *gauḥ* (a cow). He mentions and rejects *gauḥ* to be substance (*dravya*), action (*kriyā*), quality (*guṇa*) and generic property (*ākṛti*).¹⁴ Finally, he characterizes expressions like *gauḥ* to be "that, which when uttered brings forth an understanding of an object with dewlap, tail,

¹¹ एवं तर्हि निवासत आचारतश्च। स चाचार आर्यावर्त्त एव ... एतस्मिन्नार्यनिवासे ये ब्राह्मणाः कुम्भीधान्या अलोलुपा अगृह्य-
माणकारणाः किञ्चिदन्तरेण कस्याश्चिद्विद्यायाः पारगास्तत्रभवन्तः शिष्टाः। (Transl. Cardona 1997 p. 552). For a detailed discussion about the *śiṣṭas*, see also (Aklujkar 2004 p. 713-717).

¹² For details see (Joshi and Roodbergen 1986 p. 70-78).

¹³ Patañjali tells here the story of Bṛhaspati and Indra who started in this manner and gave up owing to the large numbers. If it was impossible for the divine beings, then even more so for the mortal human beings, see (P. 50-51) (Joshi and Roodbergen 1986 p. 74-75).

¹⁴ See (Joshi and Roodbergen 1986 p. 12-17) and (Ganeri 1995 p. 4-10).

hump, hoofs and horns".¹⁵ This definition by Patañjali points towards an important distinction between the form and content of any linguistic expression.

The form of an expression is its phonetic part, which is uttered and which, when heard, produces an understanding of its content or meaning.

Pāṇini and the Pāṇinīyas distinguish between the form of an expression and its content. The term commonly used for the form is *rūpa* and for the content or meaning, *artha* is employed.¹⁶ The distinction between the phonetic form of a grammatical component and its content is explicitly stated in the rule: *svaṃ rūpaṃ śabdasyāśābdasamjñā*¹⁷. Here, Pāṇini specifies that "[A] word (in a grammatical rule) which is not a technical term denotes its own form".¹⁸ For example, if a component like *agni* is mentioned in the grammar, the grammatical operations are applied only to the own form (*svaṃ rūpaṃ*) of *agni*, namely the phoneme strings / *agni* / and neither to the actual thing meant by it, nor to any other linguistic form, like *pāvaka*, that can be used for the thing meant. This is not the case for technical terms occurring in grammar. The operations are applied not on the phonetic form of the terms themselves, but on the actual constituent components for which they stand. For example, a mention of *niṣṭhā* implies the components (k)ta and (k)tavat(u).¹⁹ In a rule like *sphāyāḥ sphī niṣṭhāyām*²⁰ where the substitution is to be applied on the condition that *niṣṭhā* follows, what is meant is that if the suffixes (k)ta or (k)tavat(u) follow and not the phoneme string / *niṣṭhā* /.

Similarly, in the rule: *sarūpāṇamekaśeṣa ekavibhaktau*²¹ Pāṇini provides for the remainder of only one among those which have the same form. As explained in *Kāśīkāvṛtti*, it is important here to differentiate between the form and its content. The rule applies only when the form is identical, and not when the content is same and form may be different.²²

¹⁵ कस्त्विह शब्दः । येन उच्चारितेन सास्त्रालाङ्गलककुदखुरविषाणिनां सम्प्रत्ययो भवति स शब्दः । (P. 13-14).

¹⁶ The expression *rūpa* in the sense of form is used more than once in *Aṣṭādhyāyī* including 1.1.68, 1.2.64, 2.2.27, 3.1.94, 6.1.94. The expression *artha* for meaning or content is employed more frequently (around 80 times). Some examples are: 2.3.46, 1.1.19, 1.4.19, 1.4.85 etc.

¹⁷ स्वं रूपं शब्दस्याशब्दसंज्ञा ॥ १.१.६८ ॥ ► an expression which is not a technical term denotes its own form.

¹⁸ Transl. John Brough (1951) reprinted in (Staal 1972 p. 403). For a discussion on this rule see (Brough 1951) reprinted in (Staal 1972 p. 402-414), (Scharfe 1971 p. 40) and (Staal 1975 p. 331ff.) and more recently (Scharfe 2009 p. 182-196).

¹⁹ क्तवत् निष्ठा ॥ १.१.२६ ॥ ► (k)ta and (k)tavat(u) are *niṣṭhā*.

²⁰ स्फायः स्फी निष्ठायाम् ॥ ६.१.२२ ॥ ► *sphāy(ī)* is replaced by *sphī* before *niṣṭhā*.

²¹ सरूपाणमेकशेष एकविविभक्तौ ॥ १.२.६४ ॥ ► of those with same form, only one remains, in case of single *vibhakti*.

²² सरूपाणाम् इति किम्? वृक्ष-न्यग्रोधाः । KV on 1.2.64. Why it is stated that "having the same form"? To exclude cases like *Plakṣa* and *Nyagrodha*, which have the same meaning but different form.

In other words, the form of a technical term does not contribute to the formation of intended expression, but only its content, namely the components to which it refers. In the case of a grammatical component, however, both—its form as well as content—become important and any mention of it in the grammatical corpus refers to both. When, however, only content or meaning needs to be stated, then the expression *iti* is used after it to indicate that in this case, not the form but only its content is to be taken into consideration.²³ For example in the rule: *tasminnitinirdiṣṭe pūrvasya*²⁴ the expression *iti* after *tasmin* indicates the locative case meant by it and not the phonetic form / *tasmin* /.

The expression *artha* is used frequently by Pāṇini to denote content or meaning of grammatical elements. One of the conditions for a component to be *prātipadika* or nominal stem is that it should be *arthavat* or “having some meaning”.²⁵ He also employs the expression *artha* when the content of some component is important for grammatical operations. As an example: *matvarthe*²⁶ or *ṛtīyārthe*²⁷ when one wants to express the meaning conveyed by the suffix *mat(u)(p)* or the content of *ṛtīyā*.

How is the form of any expression related to its content? According to Kātyāyana : “there is an established (*siddha*) relation between a linguistic expression (*śabda*) and its meaning (*artha*)”.²⁸ From this, it follows that the relation between the form of any linguistic expression and its content is “established”. Further, the fixed nature of this relation is “on the basis of

Incidentally, the two trees mentioned here *Plakṣa* (*Ficus infectoria*) and *Nyagrodha* (banyan, *Ficus benghalensis* L.) belong to the same family of fig trees, but are not same.

²³ Kāśikāvṛtti on न वेति विभाषा ॥ १.१.४४ ॥ : इतिकरणोऽर्थनिर्देशार्थः । The employment of *iti* is to denote the meaning.

²⁴ तस्मिन्नितिनिर्दिष्टे पूर्वस्य ॥ १.१.६६ ॥ ► locative case indicates that the grammatical operation is to be applied to the immediately preceding component.

²⁵ अर्थवद्धातुरप्रत्ययः प्रातिपदिकम् ॥ १.२.४५ ॥ ► a meaningful component which is not a dhātu and not a pratyaya is *prātipadika*.

²⁶ तसौ मत्वर्थे ॥ १.४.१९ ॥ ► components ending in t or s before a pratyaya and having the meaning of *mat(u)(p)* are called *bha*.

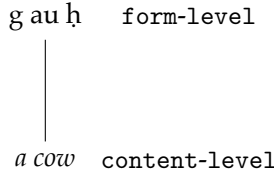
²⁷ तृतीयार्थे ॥ १.४.८५ ॥ ► *anu* is assigned *karmapravacanīya* when it denotes the meaning conveyed by *ṛtīyā*.

²⁸ सिद्धे शब्दार्थसम्बन्धे । (PV. 3). My translation of *siddha* as ‘established’ is to point to the established nature of usage of words in connection with specific meanings. Patañjali discusses the use of the expression *siddha* and suggests that what is meant is *nitya* (permanent or eternal) and clarifies that *nitya* here is not necessarily in the sense of something which is unchangeable (*kūṭastha*) or immovable (*avicalin*), but rather something that becomes established through continuous repetition (*ābhikṣṇya*) (P. 63-71). He further tries to justify the *nitya* (permanent) character of this relation and sums up by pointing towards the irrelevance of this discussion in the present context of mentioning the purpose and need of grammatical instructions (P. 72-79). For Grammarians’ discussion with the Vaiśeṣikas on the relation between *śabda* and *artha*, see (Houben 1992).

its usage among the people”.²⁹ The point is explained by Patañjali in the following words:

People, in their common usage, after having brought to mind respective meanings, accordingly use linguistic expressions. They do not make any extra effort to make or generate these expressions (from their meanings). On the other hand, they do put in extra effort to make things which are to be produced. For instance, one who needs a pot for some purpose, goes to the house of a potter and says: “You make a pot. I need a pot for some purpose”. One who wants to use linguistic expressions does not go to the house of a grammarian and says: “You make linguistic expressions. I want to use them”. He uses it according to the meaning he wants to express.³⁰

Acknowledging the distinction between the form and content of any expression, I propose it as the basis for differentiating two levels in the grammatical system: the form-level and the content-level. Any linguistic expression like *gauḥ* has some phonetic form / g au ḥ / and content *gauḥ* or *a cow*. This can be depicted as in the following figure.



Here the form-level is depicted above the content-level. The connection between the phonetic form / g au ḥ / and its content *a cow* is established through usage and is represented by the connecting line.

The above observations can be formulated in general terms as follows. Let x be any given linguistic expression. If x^f denotes its form, x^m its content and the continuous straight line the fact that their connection is established through usage, then it can be pictorially represented as in the following figure.



If the relation between a linguistic expression and its meaning is established through usage, i.e. if people are the authority with regard to these, then the

²⁹ लोकतः। (PV. 3) (Joshi and Roodbergen 1986 p. 115-116).

³⁰ यत्लोकेश्चार्थमुपादाय शब्दान्प्रयुञ्जते। न एषां निर्वृत्तौ यत्नं कुर्वन्ति। ये पुनः कार्याः भावाः निर्वृत्तौ तावत्तेषां यत्नः क्रियते। तत्पथा। घटेन कार्यं करिष्यन्कुम्भकारकुलं गत्वा आह कुरु घटम्। कार्यमनेन करिष्यामीति। न तद्वत्शब्दान् प्रयोक्ष्यमाणः वै-याकरणकुलं गत्वा आह। कुरु शब्दान्। प्रयोक्ष्ये इति। तावत्येवार्थमुपादाय शब्दान्प्रयुञ्जते। (P. 81). Translation (with minor adaptations) from (Joshi and Roodbergen 1986 p. 115-116) which includes a detailed discussion. See also (Scharf 1995 p. 74) for a slightly variant interpretation.

obvious question arises as to what then is the function of grammar. This is exactly the next question raised by Patañjali :

If the people are the authority with regard to these (i.e. the relation between meaning and the corresponding linguistic expressions), then what is the use of grammar?³¹

Kātyāyana provides an answer by introducing the concept of constraint or *niyama*.

Given that linguistic expressions are employed in accordance with their meaning fixed on the basis of its usage among the people, constraint (*niyama*) is specified through grammatical instructions for the sake of *dharma*. And this is similar to the specifications of the constraints in case of instructions about non-ritual as well as Vedic ritual actions.³²

The general idea can be stated as follows: given a number of possible options, *niyama* is constraint or restriction which can be applied to distinguish and select certain specific options out of the various possibilities.³³ In this context the following examples by Patañjali from both the Vedic as well as non-Vedic realm are illustrative and worth reproducing in full detail.

With regard to common matters (*loka*) it is said that one should not eat tame cock or tame pig. But what is food (*bhakṣya*) is taken to satisfy hunger and following this it could also be possible to satisfy hunger by eating the meat of dog etc. With regard to this, a restriction (*niyama*) is made, namely, this is eatable and this is not eatable.³⁴

Similarly, desire for a woman is because of sexual urge. Satisfaction of sexual urge is possible equally with a woman suitable for intercourse or not. With regard to this, a restriction (*niyama*) is made, namely, this is suitable and this is not suitable.³⁵

In Vedic instructions as well, it is said that a brahmin observes the vow (*vrata*) of living on milk, a *kṣatriya* on gruel and a *vaiśya* on indian cottage cheese (*āmiḥṣā*). A vow however is for the sake of taking food. It is possible to observe the vow by using rice and meat as well. With regard to this, a restriction (*niyama*) is made.³⁶

Similarly, it is said that the post for tying the sacrificial animals should be either of the *bilva* wood or of the *khadira* wood. Such a post is for the sake of tying the sacrificial animals. It is possible to tie the animal with any wooden post, standing or lying. With regard to this, a restriction (*niyama*) is made.³⁷

³¹ यदि तर्हि लोकः एषु प्रमाणं किं शास्त्रेण क्रियते। (P. 82). See also (Joshi and Roodbergen 1986 p. 117).

³² लोकतोऽर्थप्रयुक्ते शब्दप्रयोगे शास्त्रेण धर्मनियमः। यथा लौकिकवैदिकेषु। (PV. 4-5). For a detailed discussion on this topic, see (Aklujkar 2004 p. 687-732).

³³ Paul Thieme (1931 p. 23-32) discusses this point on the basis of the beginning *sūtras* of Vājasaneyi-prātiśākhya.

³⁴ लोके तावत् अभक्ष्यः ग्राम्यकुक्कुटः अभक्ष्यः ग्राम्यशूकरः इति उच्यते। भक्ष्यं च नाम क्षुत्प्रतीघातार्थम् उपादीयते। शक्यं च अनेन श्रमांसादिभिः अपि क्षुत्प्रतिहन्तुम्। तत्र नियमः क्रियते। इदं भक्ष्यम्। इदमभक्ष्यम् इति। (P. 84).

³⁵ तथा खेदात् स्त्रीषु प्रवृत्तिः भवति। समानः च खेदविगमः गम्यायां च अगम्यायां च। तत्र नियमः क्रियते। इयं गम्या इयमगम्या इति। (P. 84).

³⁶ वेदे खलु अपि पयोव्रतः ब्राह्मणः यवाग्नव्रतः राजन्यः आमिक्षाव्रतः वैश्यः इति उच्यते। व्रतं च नाम अभ्यवहारार्थम् उपादीयते। शक्यं च अनेन शालिमांसादीनि अपि व्रतयितुम्। तत्र नियमः क्रियते। (P. 85).

³⁷ तथा बैल्वः खादिरः वा यूपः स्यात् इति उच्यते। यूपः च नाम पश्वनुबन्धार्थम् उपादीयते। शक्यं च अनेन किञ्चिदेव काष्ठम् उच्छ्रित्य अनुच्छ्रित्य वा पशुः अनुबन्धुम्। तत्र नियमः क्रियते। (P. 85).

Similarly, when the potsherds have been put near the fire, the Vedic mantra : *bhrgūnām aṅgirasāṃ gharṁsya tapasā tapyādhvamiti* [be you heated with the heat of the sweat of the Bhṛgu's and the Aṅgirasas] is recited. Even without the formula, the fire, whose nature is to burn, heats the potsherds. And with regard to that, a restriction (*niyama*) is made, namely, when it is being done in this way, it leads to bliss in the form of heaven (*abhyudaya*).³⁸

In the same way here also, when meaning can be understood equally from the standard expressions (*śabda*) and non standard expressions (*apaśabda*), a restriction (*niyama*) is made for *dharma*, namely, that meaning is to be conveyed by standard expressions only and not by non-standard expressions, as usage in this manner leads to bliss in the form of heaven (*abhyudaya*).³⁹

The above quote validates the assertion that the main aim of the ancillary disciplines, including grammar, is to preserve the given standard phenomena. The given phenomena may be the admissible objects for eating, or relationships with others, the way some ritual action is performed, or some linguistic expression is uttered. The systematic approach of these disciplines is not aimed towards generating rule based *constructs*, but to place constraints in order to select some preferable possibilities, from among the several existing options.

For example, in order to express *a cow* several expressions like *gauḥ*, *gāvī*, *gonī*, *gotā*, *gopotalikā* etc. are possible. From among them, only *gauḥ* is according to the usage of model speakers, although as Kaiyaṭa mentions that “in common usage, non-standard expressions (*apaśabda*) are also used and convey the same meaning as the standard expressions (*sādhu-śabda*)”.⁴⁰

To summarize: The main task of a grammatical system is to ensure the retention of the standard usage of language. In other words, retention of the collection of the standard linguistic expressions. it is attempted to ensure this by placing constraints so that the inadmissible expressions are excluded and only the admissible ones are included.

³⁸ तथा अग्नौ कपालानि अधिश्रित्य अभिमन्त्रयते। भृगूणाम् अङ्गिरसां घर्मस्य तपसा तप्यध्वम् इति। अन्तरेण अपि मन्त्रम् अग्निः दहनकर्मा कपालानि सन्तापयति। तत्र नियमः क्रियते। एवं क्रियमाणम् अभ्युदयकारि भवति इति। (P. 85).

³⁹ एवमिहापि समानायामर्थगतौ शब्देन च अपशब्देन च धर्मनियमः क्रियते। शब्देन एव अर्थः अभिधेयः न अपशब्देन इति। एवं क्रियमाणम् अभ्युदयकारि भवति इति। (P. 86).

⁴⁰ गौरित्यस्य शब्दस्य गावी-गोणी-गोता-गोपोतलिकादयोऽपभ्रंशाः। (PB. 49). अपशब्दो हि लोके प्रयुज्यते, साधुशब्दसमानार्थश्च। (MBP. 3.1.8).

2.2 Grammatical synthesis

It is time now to focus on the rule-based process of synthesis which we depicted by the following diagram.

$$p_i \xrightarrow{\text{synthesis}} x$$

Here p_i represent the components gained by analyzing the linguistic expression x . Examples of the components include:

$$\text{bālaka, s(u), paṭh(a), (ś)a(p), ti(p).}$$

The sub-script i in p_i is the general variable which stands for natural numbers. Thus, one can refer to the above five components as:

$$p_1 : \text{bālaka, } p_2 : \text{s(u) etc.}$$

The task of grammar would be first to stipulate them and then to provide for their appropriate combinations. The combination must result in the given expression. A simple process of combination is the conjoining of two components one after another. More complex processes involve replacement, elision, augmentation and reduplication.

If Σp_i represents the combination of the constituent components, then the process of synthesis can be depicted as follows:

$$\Sigma p_i \longrightarrow x$$

The above figure says that, apart from the constituents p_i of any expression x , the grammar also provides rules of their combinations i.e. Σ which when applied to the components, leads to the standard expression x .

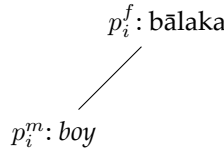
Thus, the task of grammar is two-fold: firstly to enunciate the components p_i and secondly to provide for their combinations Σ so that the combined components result in the standard expression x .

2.2.1 Form and content of a component

Like linguistic expressions, the constituent components p_i also have a form and some content. The form of any component consists of a sequence of one or more sounds. The next requirement is the specification of their content. This, I propose, can be comprehended through three basic categories: (i) the lexical meaning of a component (ii) the meaning-expressions that

are associated with a component in the grammatical corpus, and (iii) the characterizing attributes attached to the components.

The lexical meaning is the semantic content inherently associated with a component. By inherent association, I mean that it is not explicitly mentioned in the grammatical corpus. For example, the component *bālaka* means *boy* and it is assumed that the user of grammar is familiar with the meaning *boy* that is connected with the phonetic form *bālaka*. If we represent the phonetic form of any component p_i by the symbol p_i^f and the lexical content inherently associated with it by p_i^m , then this established association can be depicted as follows:



Here, the nature of association is similar to that between the form of any linguistic expression x^f and its content x^m , namely established by usage. It is therefore depicted by an unbroken line.

Pāṇinian definition of nominal stems reflects this inherent association. Nominal stems are defined as those components that have an inherent lexical meaning—and not belonging to the collection of verbal roots and suffixes.⁴¹ Pāṇini does not provide an exhaustive enumeration of the nominal stems. Instead he relies on the complement of finite sets of verbal roots and suffixes—which are enunciated in the grammatical corpus—and the condition that it must have lexical meaning.

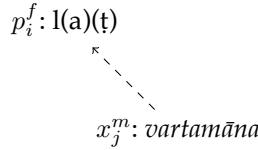
Not all components have inherent lexical meaning and in many cases grammar explicitly associates meaning-expressions to components. In the grammatical corpus, this association is usually in terms of the condition for introducing a component in the derivational process. For example, if the intention (*vivakṣā*) of the speaker is to express present time, then the component l(a)(t) should be introduced.⁴² In other words, the meaning-expression *vartamāna* (present time) is associated with the component l(a)(t).

⁴¹ अर्थवदधातुरप्रत्ययः प्रातिपदिकम् ॥ १.२.४५ ॥ ► a meaningful component which is not a dhātu and not a pratyaya is prātipadika.

⁴² वर्तमाने लट् ॥ ३.२.१२३ ॥ ► to express present time, introduce l(a)(t). In Vākyapadīya 1.24 Bhartṛhari speaks of meanings associated to the components (*apoddhāra-padārtha*). These are not fixed but there are differences of opinion (*vikalpa*) as to which meaning-expression be associated with which component. In this sense, they are different from established and fixed meanings (*sthita-lakṣaṇa*) of words. See (Rau 2002 p. 7) and (Cardona 1975 p. 280).

The user of Pāṇinian grammar must be able to correlate her or his intention with the corresponding meaning-expression in the grammatical corpus and then decide whether she or he agrees with it or not. If, for example, the speaker wants to express present time, then *vartamāna* is the corresponding meaning-expression. If the user agrees with it, then the component $l(a)(t)$ can be introduced. This is how components are associated with meaning-expressions. The component $l(a)(t)$ now carries the information that it is introduced when the user intends to express present time. Another example of this type of association is during the introduction of verbal roots. These are listed in the Dhātupāṭha with meaning-expressions that specify when a root is introduced.⁴³ A root like *bhū* would be introduced if the user understands *sattā* (existence, being) and is certain that this is what she or he intends to express. As we will see in the following pages, Pāṇini makes extensive use of meaning-expressions in his grammar.⁴⁴

The meaning-expressions are represented by the symbol x_j^m . The letter x indicates that it is a Sanskrit linguistic expression, the sub-script j refers to any of the several such expressions in the grammatical corpus. Further, the super-script m indicates that it is the meaning or content of this expression which is relevant for grammar and not its form. The association of meaning-expressions x_j^m with the form of components p_i^f can now be depicted as follows:



The dashed arrow from x_j^m to p_i^f represents the introduction of the component having the form p_i^f if the meaning-expression x_j^m is intended to be expressed.

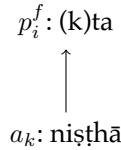
The third category is that of characterizing attributes which encapsulate grammatical or semantical information. These are designated through

⁴³ There is a divergence of opinion among the scholars on the issue whether the meaning entries formed a part of the original corpus of the Aṣṭādhyāyī as suggested by Johannes Bronkhorst (1981b p. 335-357) or were added later as maintained by Bruno Liebich (1919 p. 47-53) and G. B. Palsule (1961 p. 91ff.).

⁴⁴ Paul Kiparsky and Frits Staal (1969 p. 84) use the term “semantic representations”, Johannes Bronkhorst (1979 p. 150) uses the expression “semantic elements” and J.E.M. Houben (1999 p. 23-54) employs the phrase “meaning statements”. My choice of the term is to emphasize their semantic character as well as the fact that these are not parts or components of the whole (hence “expression” instead of “element”).

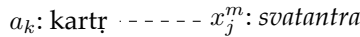
several technical terms or *sañjñās*.⁴⁵ An attribute can embody some grammatical aspect such as belongingness to a specific group of components. For example, Pāṇini groups the suffixes (k)ta and (k)tavat(u) together and calls it *niṣṭhā*.⁴⁶ In other words, he attaches the attribute *niṣṭhā* to these components.

An attribute is represented by the symbol a_k . The sub-script k indicates any of the several attributes that are specified in the grammatical corpus. The process of attachment of any attribute a_k to the form of any component p_i^f can be represented as follows:



The unbroken arrow in the figure above indicates the process of attachment of an attribute to some component.

Attributes not only comprehend some structural or systematic characteristics, but in several cases they can also represent semantic aspects. In such cases, meaning-expressions are used to specify them. For example, the attribute *karṭṛ* (agent) is defined in terms of the meaning-expression *svatantraḥ* or “that which is independent in performing an action”.⁴⁷ There is, thus, an association of a meaning-expression x_j^m with the attribute a_k . This association, however, is deliberately made in the grammatical corpus.⁴⁸ It is therefore represented through a dashed line. The following figure depicts this association.



The process of association of some meaning-expression to an attribute is different from that of attaching an attribute to some component. In the above example, the first process would be association of meaning-expression *svatantra* with the attribute *karṭṛ* and secondly, the attachment of the attribute *karṭṛ* to

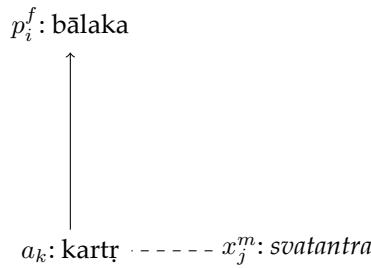
⁴⁵ There are numerous studies that deal with the Pāṇinian technical terms particularly (Renou 1942), (Chatterji 1964), (Cardona 1970 p. 195-212), (Wezler 1976 p. 361-379), (Dvivedi 1978) and (Singh 1979 p. 7-16).

⁴⁶ कर्तृवत् निष्ठा ॥ १.१.२६ ॥ ► (k)ta and (k)tavat(u) are *niṣṭhā*.

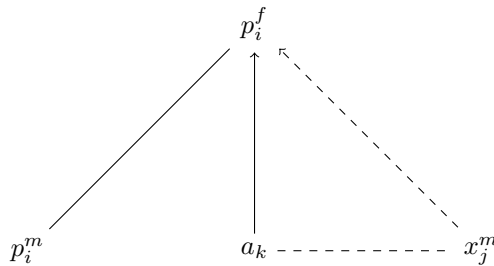
⁴⁷ स्वतन्त्रः कर्त्ता ॥ १.४.५४ ॥ ► that *kāraka* which is *svatantra* or independent of others is *karṭṛ*. See (Joshi and Roodbergen 1975 p. xviii-ix) for association of meanings or external information to the *kāraka* terms.

⁴⁸ In the words of S. D. Joshi (2001 p. 156-157): “Pāṇini establishes links between grammatical notions and non-linguistic reality ...” Joshi sees the meaning-expressions as “semantic metalinguistic statements” which “are considered to be a device to assign interpretation to the linguistic structure”.

the appropriate component, for example, the nominal stem *bālaka*. The two processes combined together can be represented as follows:



The three possible categories at the content level of any constituent component can be represented collectively as follows:



The above figure depicts the general structure of any constituent component.

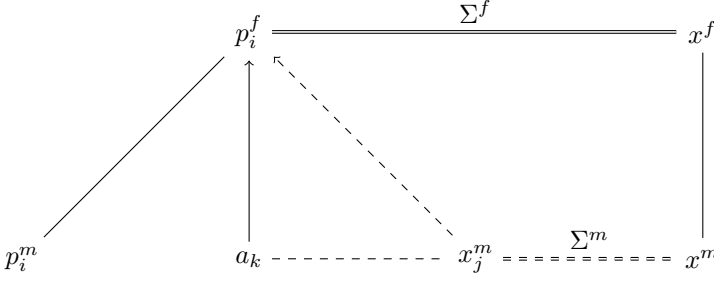
1. The two levels are distinguished as the upper form level and the lower content level. A component p_i has accordingly some form, which consists of a sequence of sounds and is depicted through the super-script f .
2. Further, it may have a lexical meaning which is not deliberately specified in the grammatical corpus. The phonetic form of lexical meaning is clearly the form of the component. This relation is similar to the one between the form and content of any given expression. These relations are established through usage. Grammar makes use of this connection, but it does not establish them. This lexical content is represented at the content level by the symbol p_i^m with the super-script m denoting the lexical meaning or content of the component. The inherent or established nature of the relationship between p_i^f and p_i^m is depicted through the unbroken line.
3. Sometimes a component is introduced by specifying certain semantic conditions. These meaning-expressions that form a condition for introduction of a component are represented by the symbol x_j^m . The dashed arrow stands for the fact that these meaning-expressions are deliberately mentioned in the grammatical corpus and form the condition for introduction of some component. The form of meaning-expression is of no direct

relevance to the process of derivation. What is of relevance is the content that it represents. In many cases it needs to be completed with other terms. Meaning-expressions in the grammatical corpus are, so to say, indices that refer to some content a user is supposed to understand and make use of. Their understanding depends upon the user's knowledge of the language, grammar and the external world.

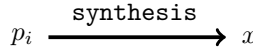
4. The next category is that of attributes a_k or names that are attached to any component. The attachment process is represented by an unbroken arrow and is specified in a deliberate manner with the grammatical corpus. The fact that attributes are placed only at the content level indicates that their phonetic form is irrelevant for the process of derivation. Formally speaking, what is important is to have a unique index through which they can be referred to. What phonetic form this index has, is of no relevance for the grammatical process.
5. Sometimes, the attributes are defined through meaning-expressions. This process is equivalent to associating a meaning-expression x_j^m with an attribute a_k which is depicted by dashed lines.

The constituent components within the grammatical system of Pāṇini can be represented in the above manner. It should be noted that the above representation makes use of the basic concept of two levels. At the form level the physical, audible sounds are represented and their content is represented through three basic categories. Further, four types of processes, depicted through four different kinds of connecting lines or arrows, are identified. Among them, only three i.e. (i) introduction of p_i^f based on x_j^m (ii) attachment of a_k to p_i^f and (iii) association of x_j^m to a_k are specified in the grammatical corpus. The lexical meaning i.e. connection of p_i^m with the p_i^f is not mentioned in it, but is taken to be established.

As mentioned before, the prime aim of grammatical synthesis is to combine the constituent components in a rule-based manner, so that the form of the combination results in the form of the desired standard expression and the sum total of the content of the components corresponds to the content of the desired standard expression. If we represent the combination of components by the symbol Σ and distinguish the combination at the form level Σ^f and content level Σ^m then the grammatical process of synthesis can be depicted as follows:

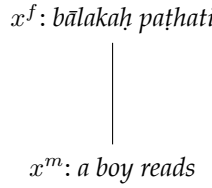


The above figure is a more detailed version of the following figure which represents that the synthesis is rule-based combination Σ of components p_i that results in x .



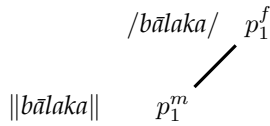
The next task is to explain how Σ , or the process of combination, functions. In order to show this, I use an example to work out the grammatical synthesis and present it in terms of the categories and processes introduced above.

1. Consider the Sanskrit expression *bālakah paṭhati* (*a boy reads*) which is intended to be derived. We have:



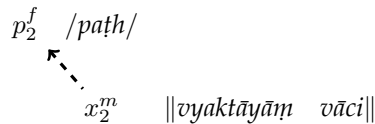
Here, the connection between the phonetic form x^f and the meaning x^m is established by usage. In other words, even without following the grammatical process of derivation, someone conversant in Sanskrit would understand what is meant by this expression.

2. In order to derive this expression one must start by selecting the appropriate constituent components that may be associated with the intended expression.
 - a. The first selection would be the component p_1 : *bālaka* whose lexical meaning is *a boy*. This amounts to saying that since the speaker intends to express p_1^m : $\|bālaka\|$ or $\|boy\|$ the component with phonetic form p_1^f : $/bālaka/$ is introduced. This is represented as follows:



It should be noted that the phonetic form of the component chosen is conducive to the phonetic form of the intended expression. Thus, components like *mānavaka* or *bāla* are not chosen, although they also convey the same meaning.

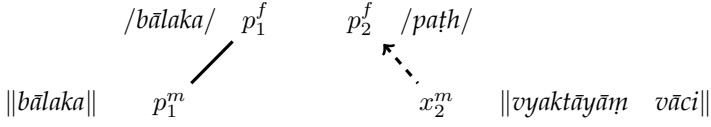
- b. The second component selected is p_2 : *paṭh(a)*. In the Dhātupāṭha, the verbal roots are listed along with the meaning-expressions that condition their introduction. Thus, the verbal root *paṭh(a)* along with the meaning-expression *vyaktāyāṃ vāci* implies that *paṭh(a)* is introduced when “expressed speech” is intended to be communicated. Thus, in this case, the component with the phonetic form p_2^f : /p a ṭh/ is specified together with the meaning-expression x_2^m : $\|\text{vyaktāyāṃ vāci}\|$ or *in case expressed speech is intended to be said*. This introduction is therefore based on some meaning-expression and can be represented as follows:



Again it should be mentioned that other components like *lap(a)* that are also listed with the same meaning-expression are not selected to avoid formation of *lapati* instead of *paṭhati*.

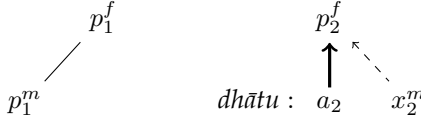
Thus, the selection of the components requires prior knowledge about them, what they denote, their lexical meaning as well as the meaning associated in the grammatical corpus. Both their form as well as their content are to be taken into consideration. Further, the two components p_1 and p_2 need to be placed in this sequence, otherwise the expression *paṭhati bālakaḥ* may result instead of *bālakaḥ paṭhati*, which has the same meaning but a different form owing to different sequencing.

The process of derivation can be considered as taking place through a series of changing derivational-states. At each derivational state a number of components, together with their contents are placed in a specific order. The derivational state after the introduction of p_1 and p_2 looks as follows:

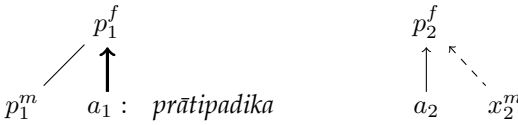


3. At this stage, the process of characterization follows. It involves attachment of a number of attributes to the components, depending upon fulfilment of appropriate conditions.

- a. The grammatical corpus consists of a table of verbal roots or the Dhātupāṭha. By the rule *bhūvādayo dhātavaḥ*⁴⁹ Pāṇini refers to it as “components beginning with bhū” and terms the components it contains as dhātu. Since the component paṭh(a) is listed in the Dhātupāṭha, therefore, an attribute a_2 : dhātu is attached to p_2^f : paṭh(a). This process can be represented as follows:



- b. Next, the first component bālaka is characterized as a nominal stem. The Pāṇinian term for nominal stems is prātipadika. As mentioned previously, unlike the verbal roots, there is no extra list of nominal stems in the grammatical corpus. Instead, they are specified by complementing the set of verbal roots and suffixes, provided they have some lexical meaning (*arthavat*). This is specified by the rule *arthavadadhāturapratyayaḥ prātipadikam*⁵⁰. Thus, the attachment of the attribute a_1 : prātipadika to the component p_1^f can be depicted as follows:

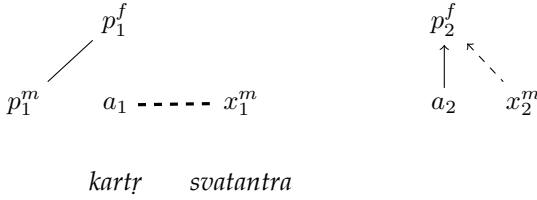


The attachment of the above attribute would include checking whether the concerned component has a lexical meaning and that it is not a verbal root or suffix.

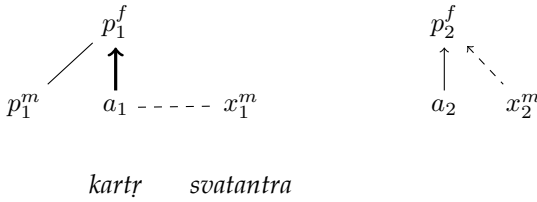
⁴⁹ भूवादयो धातवः ॥ १.३.१ ॥ ► components in the list beginning with bhū are dhātu.

⁵⁰ अर्थवद्धातुरप्रत्ययः प्रातिपदिकम् ॥ १.३.५५ ॥ ► a meaningful component which is not a dhātu and not a pratyaya is prātipadika.

- c. The component *bālaka* is also characterized as an agent. The Pāṇinian term for agent is *karṭṛ*. It is defined by the rule *svatantraḥ karṭṭā*⁵¹. The effect of this rule can be represented as the dashed line associating the meaning-expression x_1^m with the attribute a_1 . Here, to avoid further complex notations, I am using the same variable a_1 where the suffix only denotes that it is related to the first component p_1 and not the number of different attributes.



Further, the attribute *karṭṛ* or agent would be attached to the components *bālaka* if this is the intention of the speaker.



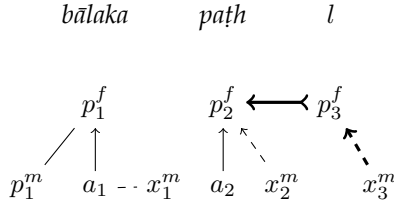
In this manner, further attributes would be attached to the components at a particular derivational state until it arrives at a point of saturation, in the sense that no more attributes can be assigned to it. These processes of attaching attributes are therefore meant to saturate the current derivational stage.

4. Once a given derivational state reaches saturation, new components can be introduced to it, depending upon the fulfilment of grammatical conditions and the intention of the speaker. The next component which is added is the suffix $l(a)(t)$. It is introduced when it is intended to express the present time.⁵² A further decision which is to be taken is whether active voice is intended, and if this is the case, then whether the suffix $l(a)(t)$ which is being introduced expresses agency.⁵³ Moreover, as a suffix it must be placed after the verbal stem. This can be depicted as follows:

⁵¹ स्वतन्त्रः कर्त्ता ॥ १.४.५४ ॥ ► that *kāraka* which is *svatantra* or independent of others is *karṭṛ*.

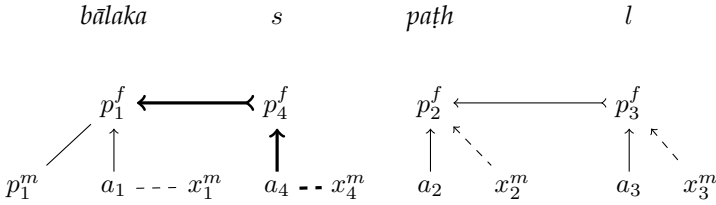
⁵² वर्तमाने लट् ॥ ३.२.१२३ ॥ ► to express present time, introduce $l(a)(t)$.

⁵³ लः कर्मणि च भावे चाकर्मकेभ्यः ॥ ३.४.६९ ॥ ► *lakāra* are used to denote the object or *karman* and the agent or *karṭṛ* in the case of transitive verbs and after intransitive verbs they denote the action or *bhāva* as well as the agent or *karṭṛ*.



There are two steps involved here. First, introduction of the component p_3^f on the basis of the meaning-expression x_3^m . This is depicted by the dashed arrow. The second step is placement of this component after the verbal root or the component p_2^f , which is specified by the arrow with a tail.

5. The next suffix $s(u)$ is introduced after the nominal stem *bālaka* and denotes nominative singular case ending. For this, it must first be decided whether the suffix $l(a)(t)$ associated with the verb expresses the agency. This implies that the attribute *kartṛ* or agent associated with the nominal stem is already expressed and therefore the first case ending can be introduced to it.⁵⁴ In case, singularity is intended, the singular case ending $s(u)$ is selected and introduced.



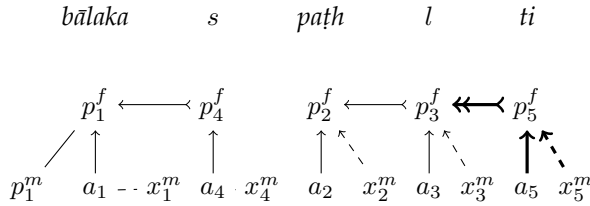
6. The following step involves introduction of the finite verb ending which replaces the suffix $l(a)(t)$ after the verb. The desired component is the third person singular suffix $ti(p)$. A number of decisions need to be taken and other grammatical constraints considered before its introduction. Since it is not in co-reference with a non-nominative,⁵⁵ the suffix $l(a)(t)$ can be replaced by one of the suffixes out of the *tiñ*-group.⁵⁶ Moreover, the fact that it is in co-reference with $[p_1 p_4]$ implies that the replacement must be

⁵⁴ अनभिहिते ॥२.३.१॥ ► when not otherwise expressed. प्रातिपदिकार्थलिङ्गपरिमाणवचनमात्रे प्रथमा ॥२.३.४६॥ ► to denote only the meaning of *prātipadika*, its gender, measure, and its number, *prathamā* is introduced.

⁵⁵ लटः शतृशानचावप्रथमासमानाधिकरणे ॥३.२.१२४॥ ► (\acute{s})at(\acute{r}) or (\acute{s})āna(c) are introduced after a verbal root in place of $l(a)(t)$ when the action is denoted at the current time and if $l(a)(t)$ is not coreferential with a *pada* which ends in *prathamā* or nominative case.

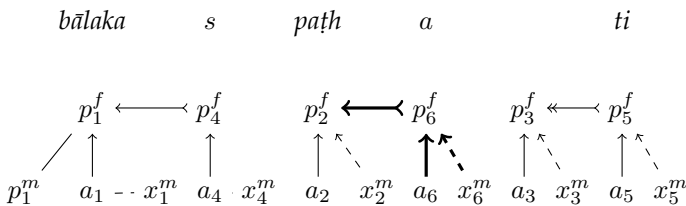
⁵⁶ लस्य ॥३.४.७७॥ ► in place of suffixes with cover term *l* namely $l(a)(t)$, $l(i)(t)$, $l(u)(t)$, $l(\acute{r})(t)$, $l(e)(t)$, $l(o)(t)$, $l(a)(\acute{n})$, $l(i)(\acute{n})$, $l(u)(\acute{n})$ and $l(\acute{r})(\acute{n})$ the suffixes coming in the next rule are substituted. तितसिञ्झसिप्थस्थमिच्वस्मस्तातांझथासाथोध्वमिङ्घिमहिङ् ॥३.४.७८॥ ► $ti(p)$ *tas jhi si(p) thas tha mi(p) vas mas ta ātām jha thās āthām dhvam i(t) vahi mahi(\acute{n})* are the substitutes of the cover term *l* or a *lakāra*.

singular. Further, *paṭh(a)* is a parasmaipada verbal root and neither first nor second person is being expressed, therefore, *ti(p)* is selected which replaces *l(a)(t)*.



In the above figure, the double arrow depicts the process of replacement.

7. In the next step, an infix (*ś*)*a(p)* is introduced after the verbal root and before the finite case ending. If the case ending suffix denotes agency—which is inherited from the suffix *l(a)(t)* which it replaced—and if it is a *sārvadhātuka* suffix⁵⁷ then the infix (*ś*)*a(p)* is introduced.



8. The next change follows the rule *sasajūṣo ruḥ*⁵⁸, with the substitution of *r(u)* in place of *s*.

b ā l a k a r p a ṭ h a t i

9. Finally, following the rule *kharavasānayorvisarjanīyah*⁵⁹ the phoneme *r* is replaced by *visarjanīya* because of the following *khar* (i.e. *p*) sound.

b ā l a k a ḥ p a ṭ h a t i

From an operational point of view, both the above replacements are similar to the previous replacement of *l(a)(t)* by *ti(p)* in step 6.

The process of derivation presented in the above example is slightly different from that suggested by Joshi and Roodbergen (1980 p. ix-xi). The main difference is that they start by associating the meaning-expressions with the

⁵⁷ This is because it belongs to the group called *tiñ*. तिङ् शित् सार्वधातुकम् ॥३.४.११३॥ ► *tiñ* and those having *ś* as its marker are called *sārvadhātuka*.

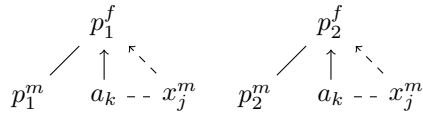
⁵⁸ ससजुषो रुः ॥८.२.६६॥ ► final *s* as well as *ś* of *sajūṣ* is replaced by *r(u)* at the end of a pada.

⁵⁹ खरवसानयोर्विसर्जनीयः ॥८.३.१५॥ ► before *khar* or pause (*avasāna*), *r* is replaced by the *visarjanīya* (= *ḥ*) provided this *r* is final in a pada.

(kāraḱa) attributes and introduce the constituent components like bālaḱa or paṭh(a) at a later stage.⁶⁰ In my representation, the attributes occur together with the constituent components in a composite manner. There is no pure or deeper semantic or syntactic level, but the components are a complex of pho-netic form together with lexical, semantic and grammatical content.

2.3 The derivational process

The application of the grammatical system to synthesize a particular expression is carried out in a number of steps. Each step can be said to correspond to some derivational state. Such a derivational state would consist of a sequence of components. Its detailed specification at the form and content level would then consist of respective categories. A general derivational state with two components is shown in the following figure.



There are two components here, p_1 and p_2 , with their respective forms p_1^f and p_2^f and lexical contents p_1^m and p_2^m . Further, a number of grammatical attributes a_k or semantic content x_j^m can be attached to them.

2.3.1 Guiding principles of synthesis

The process of synthesis involves changes in the derivational state. A change could be at the content level, for example, when new attributes are attached or at the form level when new components are introduced. Any change in a given derivational state is guided by

1. the intention (*vivakṣā*) of the speaker, and
2. the constraints of the grammatical system, which involves
 - a. consistency of the derivational state
 - b. its saturation, and
 - c. its completion.

⁶⁰ See, for example, the stages A(1) to B(4) in (Joshi and Roodbergen 1980 p. ix-x). Brendan Gillon (2007 p. 451-461) follows a similar approach.

2.3.2 Intention of the speaker

The intention of the speaker relates either to the meaning-expressions x_j^m or to the lexical meaning p_i^m of the constituent components. The grammatical system interacts with the speaker and gathers information about her or his intention through meaning-expressions. Intention is also related through the meaning of a lexical component.

The meaning-expressions are explicitly stated in the grammar. A user must be able to understand them and react to them. For example, the meaning-expression *varṭamāna* requires that the user understands what is meant by it—namely, present time—and decides whether she or he wants to express it or not. The lexical meanings, on the other hand, are not explicitly stated in the grammar. A user must be able to choose the right lexeme or component that corresponds to the lexical content she or he wants to express and input it into the grammatical system. This requires familiarity with the lexemes of the language. This is necessary because the form and content of such components are fixed or established through usage.

It does not suffice to know only the meaning. Form is equally important. For example, if one wants to express *bālakaḥ paṭhati*, then one must select *bālaka* and not *bāla* or *māṇavaka* which also mean the same. Similarly, one must choose *paṭh(a)* and not *lap(a)* although both are listed in the Dhātupāṭha in the sense of *expressed speech* (*vyaktāyām vāci*). Otherwise, *lapati* instead of *paṭhati* may result.

2.3.3 Consistency of the derivational state

The rules of grammar provide for the consistency of a derivational state. The constraint of consistency regulates the distribution of the components and the categories expressing their content. At a given state, the sequence of components, their phonetic combinations and the presence or absence of grammatical attributes are specified in the grammar.

As an example, consider the placement of suffixes (*pratyaya*). This is regulated by the rule which specifies that components with the attribute *pratyaya* must be placed after the component for which they are introduced.⁶¹ Another example of the constraint of consistency is the presence of a particular attribute in a component, which sometimes excludes the possibility

⁶¹ प्रत्ययः ॥३.१.१॥ ► *pratyaya* are components introduced subsequently. परश्च ॥३.१.२॥ ► and are placed after the components to which they are introduced.

of attaching certain other attributes to that component. Such attributes can be grouped within a set of mutually exclusive attributes. For example, one such set is {hrasva, dīrgha, pluta}. If a phoneme unit is hrasva then it can not simultaneously be dīrgha or pluta. Similarly, a component with attribute kartṛ can not simultaneously have the attributes karman, sampradāna etc.⁶²

Only a consistent derivational state is admissible.

2.3.4 Saturation of the derivational state

Given a derivational state, a number of attributes can be attached to the components. Saturation of the derivational state is reached once no further attributes can be attached at that particular state. As long as some attribute can be attached to any component of a particular derivational state, it is not saturated.

Attachment of attributes in general leads to grammatical characterization of components, as well as of the derivational state. They can be attached to the phonetic form of any component. This can be either a single sound i.e. a phoneme, or a sequence of them which may represent one or more constituent components. The first case has phoneme attributes and such attributes can be distinguished from other attributes, i.e. those that characterize phoneme sequences. Consider, for example, the component (k)ta appearing at some stage in the derivational process. The phoneme t is termed *tu*, *khay*, *khar* and a is *ac*, *at*, *guṇa* etc. On the other hand, the component (k)ta is called *niṣṭhā*, *pratyaya*, *kit* etc.

There are many attributes, which once attached to a component, do not leave it. For example, the attribute *ac* for the phoneme / a /. This means, once attached, this attribute remains associated with that component in all the subsequent states as the derivational process advances. Such characteristics can be termed as *static-attributes*. On the other hand, certain attributes are assigned temporarily to some component. They depend upon the current derivational state and once its distribution changes, they no longer remain attached to that component. One such example is the attribute *aṅga*.⁶³ This is assigned with respect to some suffix and depending upon the distribution of suffixes in a particular state, it is attached to corresponding components, but

⁶² In Aṣṭādhyāyī this is specified under *eka-sañjñā* i.e. “assignment of only one term” constraint prescribed by the rule: आ कडारात् एका संज्ञा ॥ १.४.१ ॥ ► upto a.2.2.38 only one sañjñā or attribute is attached (from among the set of mutually exclusive attributes).

⁶³ यस्मात्प्रत्ययविधिस्तदादि प्रत्ययेऽङ्गम् ॥ १.४.१३ ॥ ► that part which enjoins a pratyaya based operation, before that the sequence is aṅga.

in the subsequent state things may change as may the assigned position of this attribute.⁶⁴ Other examples include *ṭi*⁶⁵ or *upadhā*⁶⁶ that are assigned on the basis of current positions of the phonemes.

Another criterion on the basis of which attributes are distinguished is whether some meaning-expressions are required for their assignment. If this is the case, then some kind of user intervention is required to interpret them and accordingly attach the attributes to the appropriate component. Consider the assignment of the attribute *karṭṛ*, which is specified in terms of a meaning-expression *svatantra*.⁶⁷ Here, the component which denotes the independent agency of action is assigned the attribute *karṭṛ*. This involves assistance from the user. On the other hand, several attributes can be assigned only on the basis of distribution of system internal parameters and without the help of the user. The grammatical system can decide on its own. For example, the attribute *niṣṭhā* can be assigned without consulting the user, solely on the basis of the presence of (k)ta or (k)tavat(u).⁶⁸

Given a distribution of components, the derivational state must attain a state of saturation before new components can be introduced or changes can be effected at the form level.

2.3.5 Completion of the derivational state

A saturated derivational state is complete if no more components can be added to it. Neither the user, who interacts through the meaning-expressions, nor the grammatical system calls for addition of any component. Such a complete state must fulfill two conditions, namely (1) the phonetic form reached must be identical with the phonetic form of the intended linguistic expression and (2) the collective content of the completed state must correspond to the content of the intended expression.

A derivational state, if incomplete, advances towards completion. This takes place when a new component is added to it. When it is placed adjacent to some existing component then it is a case of simple addition or augmentation (*āgama*), and when it replaces some component then it is substitution

⁶⁴ James Benson (1990) discusses Patañjali's remarks on *aṅga*. H. V. Nagarajarao (1978 p. 145-176) discusses the scope and necessity of the *adhikāra* of *aṅga* although Pāṇini does not explicitly states its domain.

⁶⁵ अचोऽन्त्यादि टि ॥ १.१.६४ ॥ ► that part which begins with last ac is *ṭi*.

⁶⁶ अलोऽन्त्यात्पूर्वं उपधा ॥ १.१.६५ ॥ ► an *al* which is penultimate is *upadhā*.

⁶⁷ स्वतन्त्रः कर्त्ता ॥ १.४.५४ ॥ ► that *kāra* which is *svatantra* or independent of others is *karṭṛ*.

⁶⁸ क्तवत् निष्ठा ॥ १.१.२६ ॥ ► (k)ta and (k)tavat(u) are *niṣṭhā*.

(*ādeśa*).⁶⁹ Augmentation or substitution of phonemes leads to change in the form of a component. Replacement of the entire components, however, not only leads to change in form but facilitates deeper layers of abstraction and generalization in formulation of grammar. The best example is the introduction of *abstract* components like l(a)(t) and their complete replacement by case endings.⁷⁰ Pāṇini specifies *where* the new components should be placed. For example, the suffixes are added after the component to which they are introduced.⁷¹ Components marked with ṭ or k are added at the beginning or at the end of the constituents to which they are introduced.⁷² A component with m as marker is added after the last vowel of the component to which it is introduced.⁷³ Substitution involves a combination of two processes, namely the component which is replaced is made invisible by assigning an attribute replaced⁷⁴, and the substitute is introduced at the appropriate position.

2.3.6 Conditions

The forces that induce some change in the derivational state are specified through conditions. The process of saturation of a derivational state, or its transition towards completion from one state to the next, depends upon these conditions. They are formulated in terms of the distribution of components and their contents in the current state, in the previous state and sometimes in the subsequent future state as well. The nature of conditions and their complexities will be discussed in the following chapters 3 and 4.

⁶⁹ On the concept of substitution (*ādeśa*) and its possible origins in the Brāhmaṇa and Upaniṣad texts, see (Thieme 1968 p. 715-723). See also (Wezler 1972 p. 7-20) on *sthānin* (substituendum).

⁷⁰ For replacement techniques in Pāṇini, see (van Nooten 1967 p. 883-902). See also (Joshi and Roodbergen 1975 p. xvii-xix).

⁷¹ प्रत्ययः ॥ ३.१.१ ॥ ► pratyaya are components introduced subsequently. पश्च ॥ ३.१.२ ॥ ► and are placed after the components to which they are introduced.

⁷² आद्यन्तौ टकितौ ॥ १.१.४६ ॥ ► a ṭit element is placed at the beginning and a kit at the end of the component to which it is assigned.

⁷³ मिदचोऽन्त्यात्परः ॥ १.१.४७ ॥ ► a mit element is attached after the last ac of the component to which it is assigned.

⁷⁴ This is a new attribute which I use, instead of lopa. It represents those units which are now replaced because of substitution. In normal application of grammar, sometimes it is retained *in memory* (e.g. replacement of lakāras) and sometimes it is safely forgotten (e.g. in phoneme replacements).

Chapter 3

Frameworks

The corpus of Aṣṭādhyāyī formulates the grammatical system specified in the previous chapter in a special framework. I will mention some of the salient features of the Pāṇinian formulation in the following section. The present chapter, however, introduces a new formal framework which facilitates a *re-presentation* of the Aṣṭādhyāyī. The framework I propose is different from the one in which the Aṣṭādhyāyī is formulated.

The most significant difference between Pāṇinian and formal representation is that of the medium. Without entering into the debate about whether script was known to Pāṇini or whether he made use of it when compiling his grammar, based on the manner in which the grammar is formulated—e.g. the use of accents or phonemes as markers—it can be asserted that Aṣṭādhyāyī is composed, transmitted and applied in an oral framework. Moreover, it is composed in Sanskrit, although it employs a number of special meta-linguistic conventions. A formal representation, on the other hand, is meant to render it in terms of logical relations that can be implemented on a computer.

The reason for proposing a new framework is that the Pāṇinian formulation, although special on several counts, is not adequately formal for a direct implementation on computer. Apart from its oral and verbal character, Aṣṭādhyāyī is meant to be used by a person who has acquired skills to apply it through the study of grammar. The rules of grammar are not listed in the order in which they are to be applied for derivation of any linguistic expression. Instead, rules having common contexts are grouped together to avoid repeated and redundant enunciation of these contexts. A formal representation, however, is meant to be applied by a computer program with the aid of a user. For this, a mechanism needs to be developed by which the required information scattered in the grammatical corpus can be collected together using various indices encoded in a complex meta-language.

An important challenge in Pāṇinian formulation is the presence of ambiguities, inconsistencies and the incomplete nature of the Aṣṭādhyāyī. I will discuss some of these issues in section 3.2. While the commentaries and other explanations available in the later grammatical literature help in clarifying them, a device is still needed to incorporate these explanations and suggestions. Instead of directly modeling the corpus of Aṣṭādhyāyī, the present approach models the categories and processes outlined in the previous chapter about the grammatical system. It also facilitates incorporation of information necessary for application of the grammar but not explicitly mentioned in the grammatical corpus.

3.1 Salient features of the Pāṇinian formulation

The formulation of the grammatical system in the Aṣṭādhyāyī of Pāṇini is in the form of concise statements—the *sūtras*.¹ Typically, a *sūtra* consists of one or more inflected expressions. In the following, I shall call them elements of a *sūtra*. These elements are inflected just like any other expression in Sanskrit. For example, in a *sūtra* like *veḥ pādaviharaṇe*² the element *veḥ* is the ablative singular of *vi* and *pādaviharaṇe* is locative singular of *pādaviharaṇa*. Pāṇini's description of Sanskrit, therefore, is in Sanskrit.

Yet not all the elements within the grammatical corpus belong to the common language. There are many elements that appear only in the grammatical corpus. In the above *sūtra*, the second element *pādaviharaṇe* belongs to the common speech and the first one *veḥ* is an entity which exists only within the grammatical system. There is thus a clear distinction between the object language which is being described i.e. Sanskrit and the language of the Aṣṭādhyāyī, which is a special language of description. From Pāṇini onwards the tradition clearly recognizes two types of languages: the perennial utterances of the Vedas and the established language of everyday communication on the one hand and on the other hand the artificial language of grammatical instructions. The language of grammar is for the specific purpose of providing a description of the standard usage and is artificially created by the grammarians. In this sense, Pāṇini is not the creator (*kartr̥*) but the one who re-collects (*smart̥*) the object language.³

¹ Louis Renou (1963 p. 165-216) provides a detailed survey on the genre of *sūtra* in Sanskrit literature. See also (Staal 1992 p. 303-314) and (Wezler 2001a p. 351-366).

² वेः पादविहरणे ॥ १.३.४१ ॥ ► *ātmanepada* suffixes are attached after *kram(u)* if it comes together with the preverb *vi* to express nice strides (*pādaviharaṇa*).

³ A detailed discussion of the traditional points of view on this topic is provided by R. S. Bhattacarya (1966 p. 212-237). Hartmut Scharfe (1971 p. 2-6) also gives a summary of the main arguments.

The artificiality of the language of Aṣṭādhyāyī also lies in the fact that it follows only partially the rules of grammar that it lays down. In many cases it is guided by a special set of rules, namely the meta-linguistic conventions. For example, although the elements of the grammatical corpus consist of inflectional suffixes like any other linguistic expression, the meaning which they denote is different compared to what they express in common usage. While the rules under kāraka-section relate to common usage,⁴ in case of elements of the grammatical corpus several meta-linguistic conventions provide for other denotations. Thus, for example, the convention *ṣaṣṭhī sthāneyogā*⁵ specifies that the sixth case (ṣaṣṭhī) or the genitive case-ending denotes the place where a substitution takes place (i.e. the substituendum). The right-context is expressed through locative endings and this is specified by the meta-rule *tasmīnītinirdiṣṭe pūrvasya*⁶. Similarly, the convention *tasmādityuttarasya*⁷ specifies that the left-context is denoted through ablative endings.⁸

Considering the special meta-linguistic conventions followed in the formulation of the grammatical corpus, and the fact that several of these conventions are explicitly mentioned in the same corpus, the statements of Aṣṭādhyāyī can be grouped into two basic types:

1. Operational rules that introduce the grammatical components, characterize them and provide for their combinations.
2. Meta-linguistic conventions about the formulation of the operational rules.

The first group deals with what is to be done and the second one with how it is presented or formulated in the Aṣṭādhyāyī.

In the previous chapter, I introduced three kinds of constituents of grammar: (i) the components of linguistic expressions (ii) the attributes that characterize them and (iii) the meaning-expressions specifying the semantic information. As mentioned before, within the grammatical corpus, these constituents occur in an inflected form. The combination of the inflectional suffix with the base follows the normal rules of suffix additions and phoneme combinations

⁴ The section on kāraka is from 1.4.23-1.4.55. For a detailed discussion on the kāraka's see (Cardona 1974) and the Kārikāhnikā of the Mahābhāṣya (Joshi and Roodbergen 1975).

⁵ षष्ठी स्थानेयोगा ॥ १.१.४९ ॥ ► ṣaṣṭhī stands for 'introduction in place of'.

⁶ तस्मिन्निनिर्दिष्टे पूर्वस्य ॥ १.१.६६ ॥ ► locative case indicates that the grammatical operation is to be applied to the immediately preceding component.

⁷ तस्मादित्युत्तरस्य ॥ १.१.६७ ॥ ► ablative case indicates that the grammatical operation is to be applied to the immediately following component.

⁸ It should be noted that the meta-linguistic conventions are not universally applied in a consistent and consequent manner throughout the Aṣṭādhyāyī. See the discussion in section 3.2.

laid down in the Aṣṭādhyāyī. For example, the attribute *vṛddhi* is stated as *vṛddhiḥ* in the rule *vṛddhirādaic*⁹. Similarly, the meaning-expression *svatantraḥ* in rule *svatantraḥ karttā*¹⁰ is mentioned together with the inflectional suffix for nominative singular.

Rules of phoneme combination and compounding are applied as well. The components have a special presentational scheme. While the meaning-expressions and many (but not all) attributes are from the natural language, the components do not belong to the common usage. Their artificial character is reinforced by the use of marker-sounds that are often attached to them either at the beginning or at the end. Thus, in a component like *ti(p)*, the final sound *p* is a marker. In this book, markers are pointed out by including them within brackets (). Similarly, the component *(ś)a(p)* has a marker *ś* attached at the beginning and another marker *p* at the end.

Apart from markers, the components sometimes have a vowel which is needed just for facilitating its pronunciation. Thus, *n[u](m)* consists of the vowel *u* which is not part of the component, but is added in order to be able to pronounce this component. Such extra vowels are noted within square brackets [] in this book.

3.1.1 Methodology of Pāṇinian formulation

The elements of the grammatical corpus are threaded within a number of inter-connected *sūtras* (lit. a string). A *sūtra* constitutes a distinguishable unit of the grammatical corpus with a specific function.¹¹ Although *sūtra* is a distinguishable functional component, it is not always an independent unit and in most of the cases several *sūtras* must be read together in order to execute a grammatical operation. A significant Pāṇinian device which is employed for this purpose is the carrying over of elements from previous to subsequent rules (*anuvṛtti*).¹² Moreover, many rules operate within the domain set by what are known as *adhikāra* rules. In this manner repetitions are avoided and enunciation of an element at one place suffices for its use

⁹ वृद्धिरदैच् ॥ १.१.१ ॥ ► *vṛddhi* stands for *āt* and *aic*.

¹⁰ स्वतन्त्रः कर्त्ता ॥ १.४.५४ ॥ ► that *kāra* which is *svatantra* or independent of others is *karttā*.

¹¹ Tradition speaks of six kinds of *sūtras* based on their function: *sañjñā* (definition), *paribhāṣā* (convention), *vidhi* (injunction), *niyama* (restriction), *pratiśedha* (prohibition) and *adhikāra* (expansion). See (Abhyankar 1974 p. 432).

¹² S. D. Joshi and Saroja Bhate (1984) have comprehensively examined the principles behind such an organization of the grammatical corpus in a monograph titled "The fundamentals of *anuvṛtti*".

at other parts of the corpus.¹³ The method of *anuvṛtti* functions at the level of grammatical elements. In most of the cases it is only a part of the *sūtra* which is carried over to the subsequent rules. Consider for example the rule *kṛnmejantaḥ*¹⁴, which says that those expressions that end in *kṛt* suffixes having *m* or *ec* at the end, are termed *avyaya*. The term *avyaya* here is to be taken from the rule *svarādinipātamavyayam*¹⁵ where it is first introduced. So a complete reading would be *kṛnmejantaḥ avyayam*, where the term *avyaya* is collected from a foregoing rule. Not the entire rule, but only a part of it is carried further.¹⁶

Since application of a grammatical operation requires elements from more than one *sūtras* to be collected together, it becomes necessary to specify the boundaries of elements within the grammatical corpus. In fact, grammatical corpus can be considered as a sequence of elements where *sūtra*-boundary is marked as well.

The conciseness of a *sūtra* is firstly because as a unit it is not always a complete and self-contained expression of what is to be said, but only a part of the whole statement. It needs to be completed by incorporating other information. The information lacking may be present in some other part of the corpus and must be fetched to make a complete and applicable statement. Sometimes the missing information is to be provided through conventions and interpretations not explicitly mentioned in the grammatical corpus. A *sūtra* therefore is a unit within an inter-dependent and inter-related network.

The organization of the grammatical corpus in terms of the *sūtras* consisting of one or more elements, points towards the fact that the formulation of the grammatical corpus follows the same methodology as the grammar itself. As specified in the previous chapter, this approach is to analyze a given whole into constituent components and then provide a rule-based combination of the components to specify the whole. In the case of the corpus of grammar, the *sūtras* (or parts thereof) are components that need to be combined with other components (i.e. with other *sūtras* or parts thereof) in order to fetch an applicable complete rule of grammar. How the different components of an applicable rule are to be gathered together is guided by

¹³ H. V. Nagarajarao (1978 p. 145-176) discusses the scope and necessity of the *adhikāra* of aṅga although Pāṇini does not explicitly states its domain. The meaning-*adhikāras* in the *taddhita* section of the *Aṣṭādhyāyī* are analyzed by Saroja Bhate (1987 p. 81-92). Ashwini Deo (2007 p. 1-37) shows that the *taddhita* section of the *Aṣṭādhyāyī* is structured as a default inheritance hierarchy.

¹⁴ कृन्मेजन्तः ॥ १.१.३९ ॥ ► expressions that end with those *kṛt* suffixes which end in *m* or *ec* are *avyaya*.

¹⁵ स्वरादिनिपातमव्ययम् ॥ १.१.३७ ॥ ► *svarādi* (*svar* etc.) and *nipāta* are *avyaya*.

¹⁶ In his six volume edition of the *Aṣṭādhyāyī*, Rama Nath Sharma (1990 p. ix-x) has provided *anuvṛtti* elements from preceding *sūtras*.

a number of meta-linguistic conventions, as well as several interpretations supplied by the later Pāṇinīyas.¹⁷

As mentioned in the previous chapter, grammar comprehends the standard usage by analyzing and identifying the constituent components and then providing a rule-based combination of them. Similarly, the formulation of the grammar is achieved in terms of *sūtras* (including their constituent elements) and together with conventions to synthesize the operational statements.

The purpose of opting for the same methodology is also similar, namely to provide a compact encoding of the grammatical information as well as to safeguard it against future loss or corruption. Several techniques are employed to this effect. The most important method is to enunciate first the general characteristics and then to specify the exceptions.¹⁸ Components that share common grammatical characteristics are linked together. Two such groups are relatively large and enumerated separately. These are the list of verbal roots (Dhātupāṭha) and of selected nominal stems (Gaṇapāṭha). There are several sub-groups within these groups and they are frequently referred to in the main corpus.¹⁹ Formation of groups is attained using marker sounds as delimiters which facilitate specification of sigla (*pratyāhāra*).²⁰ The it-markers as indicators that are directly attached to components serve to form groups as well.

Meta-linguistic information is also encoded in the intonation of the components and *sūtras*. Thus, according to the convention *svaritenādhikāraḥ*²¹, the circumflex accent (svarita) indicates that the particular *sūtra* (or a part of it) is a heading rule. Similarly, the verbal roots are specified with accent markers which aid in deciding the kind of suffixes that should be attached to them. For example, following the rule *anudāttanīta ātmanepadam*²², the ātmanepada suffixes are attached if the verbal roots are marked by a low pitched (anudātta) marker vowel.

Pāṇini also uses the principle of correspondence and instead of mentioning individual pairs he correlates collectively two lists of equal length.

¹⁷ Despite copious literature on this subject the task is by no means finished. For example, there are cases where scholars have divergent opinions as to the boundary of a *sūtra*.

¹⁸ Patañjali notes this in (P. 53), see (Joshi and Roodbergen 1986 p. 77).

¹⁹ For example, the ajādi group is referred to in the rule अजाद्यतष्टाप् ॥४.१.४॥ ► after ajādi components and those ending in *at*, suffix (t)ā(p) is introduced to denote feminine. See the Gaṇapāṭha in (Katre 1989 p. 1265-1325).

²⁰ See section 4.1.1.

²¹ स्वरितेनाधिकारः ॥१.३.११॥ ► through svarita a domain is marked.

²² अनुदात्तङित आत्मनेपदम् ॥१.३.१२॥ ► after anudāttet or nit components ātmanepada suffixes are introduced.

The convention *yathāsaṃkhyamanudeśaḥ samānām*²³ specifies this. The rule *iko yaṇaci*²⁴ is an example. Here *ik* = [i, u, r, l] and *yaṇ* = [y, v, r, l] are correlated respectively.

For the specification of it-markers, Pāṇini makes use of rules.²⁵ Moreover, the names of the sets of phonemes (*pratyāhāra*) are generated in a rule-based manner. The convention *ādirantyena sahetā*²⁶ is used to coin the names of the collections in a generic manner. Thus, Pāṇini not only attains brevity by using the acronyms or *pratyāhāra* instead of using the entire list every time, but he also generates their names using just one rule.

The employment of several of the techniques mentioned above to achieve maximum compactness has the consequence that the grammatical corpus attains a complex structure. This requires commentaries and explanations to make explicit the encoded information in an unambiguous manner. The task of formalization of the grammar needs therefore to take this aspect into consideration.

The formulation of the grammatical system does not furnish in an explicit manner the process of derivation which involves operations that are carried out one after the other in a sequential manner. These operations are conditioned. To execute this process, one would ideally expect that the next applicable operation is unambiguously specified. It means that there is a unique operation and all the conditions that are to be satisfied are also available. A *sūtra*, however, is not always prescribing a unique operation and seldom does it have the complete set of conditions mentioned at one place. For example, the *sūtra*: *kartuḥ kyaṇ salopaśca*²⁷ provides for two operations: introduction of the suffix (k)ya(ṇ) and elision of the final phoneme s. For the conditions other *sūtras* like *dhātoḥ karmaṇaḥ samānakartṛkādicchāyāṇi vā*²⁸ and *supa ātmanah kyac*²⁹ etc. need to be taken into account.³⁰

²³ यथासंख्यमनुदेशः समानाम् ॥ १.३.१० ॥ ► respective assignment for equal number of elements.

²⁴ इको यणचि ॥ ६.१.७७ ॥ ► *ik* is replaced by *yaṇ* if it is followed by *ac*.

²⁵ The rules for it markers are: उपदेशेऽजनुनासिक इत् ॥ १.३.२ ॥ ► in grammatical instruction, an *ac* which is *anunāsika* is *it*. हलन्त्यम् ॥ १.३.३ ॥ ► *hal* coming at the end. न विभक्तौ तुस्माः ॥ १.३.४ ॥ ► but not in *vibhakti*, the *tu*, *s* or *m*. आदिर्ङितुडवः ॥ १.३.५ ॥ ► the initial *ṅi*, *ṭu* and *ḍu* (of *dhātu*). षः प्रत्ययस्य ॥ १.३.६ ॥ ► *ṣ* of *pratyaya*. चुटू ॥ १.३.७ ॥ ► *cu* or *ṭu*. लशक्तद्धिते ॥ १.३.८ ॥ ► *l*, *ś* and *ku* of non *taddhita*.

²⁶ आदिरन्त्येन सहैता ॥ १.१.७१ ॥ ► an initial element together with the final it sound includes intervening elements.

²⁷ कर्तुः क्यङ् सलोपश्च ॥ ३.१.११ ॥ ► after *kartṛ* (k)ya(ṇ) is introduced and the final *s* is elided (*lopa*).

²⁸ धातोः कर्मणः समानकर्तृकादिच्छायां वा ॥ ३.१.७ ॥ ► after *dhātu* optionally in case its action is *karman* of some *dhātu* expressing desire and both *dhātu* have same *kartṛ*.

²⁹ सुप आत्मनः क्यच् ॥ ३.१.८ ॥ ► after *sup* to express desire for oneself, (k)ya(c) is introduced.

³⁰ For details see (R. N. Sharma 1987 p. 46).

Here it is important to note that it is not that the grammar does not specify the conditions, but only that the specification through the *sūtra*-style is such that these are distributed at different locations and need to be gathered at the time of application. This process of bringing all the requisite information together in one place is carried out by a human user on the basis of her or his knowledge of the Aṣṭādhyāyī, especially the knowledge of its meta-linguistic conventions.

In an article published in 1975, Rama Nath Sharma (1975 p. 31-39) discusses this aspect. He suggests that there is an implicit device which works via reference to retrieve information necessary for the proper application of rules. He refers to the employment of domains and recurrences (*adhikāra* and *anuvṛtti*) as well as the use of technical terms, which necessitates the construction of what he terms as *referential indices*, so that the required information can be gathered for the application of rules. In the first volume of his six volume edition of the Aṣṭādhyāyī he elaborates this idea further providing ample examples (R. N. Sharma 1987 p. 60-73). It suffices here to mention that his suggestions are largely an attempt to note down the (mental) process through which relevant information is gathered in order to apply the rules of grammar. It is clear that in a formal representation, which a computer should be able to understand and apply, this process needs to be made explicit. Although Sharma rightly points out its need and explains how it can be done following applicational procedure (*prakriyā*) of Pāṇini, he does not provide a practical framework in which it may be realized.

In contrast to the suggestion put forward by R. N. Sharma (1975 p. 31), I propose that it is necessary to recast the content and processes of grammar in a new formal framework. The framework which I intend to introduce differs from the way in which the Aṣṭādhyāyī is formulated by Pāṇini. The prime focus of the new formal framework is to integrate the representational and applicational aspects of grammar. The rules, once recast in the new framework, can be interpreted and applied in an algorithmic manner and the derivational process can be carried out with the aid of computer systems. Another important aspect of the new framework is its non-oral or non-verbal character in a formal and unambiguous manner. This enables on the one hand the object and the meta languages to be clearly distinguished, and on the other hand facilitates computer implementation.

The proposal I make to recast the Aṣṭādhyāyī in terms of a new formal framework is significant with far reaching consequences for the organization and presentation of the grammatical corpus. An immediate question arises here: why should one attempt to represent the Aṣṭādhyāyī in a new framework? Why not keep the Pāṇinian formulation that evinces several formal features? After all, it follows a systematic and advanced meta-language which, to a large extent, is employed in an unambiguous and consistent

manner. Further, looking at the large amount of literature claiming the Aṣṭādhyāyī to be the oldest and best example of formal representation, it seems an uninvited exercise.

Despite several advantages—like safeguards against human frailty—achieved through concise, well knit, verbal aphorisms in which the Aṣṭādhyāyī is formulated, its complex meta-language requires an elaborate and comprehensive apparatus for interpretation and application of the grammatical corpus. The framework in which these are formulated is suitable for oral transmission and human application of the grammar, but not suitable for its formal representation and algorithmic application. In the following I will first discuss the problems of formalization of the Aṣṭādhyāyī, and then introduce a new formal framework to represent it.

3.2 Formalization of the Pāṇinian corpus: challenges and possibilities

There are certain practical challenges if one decides to keep to the verbal framework in which Aṣṭādhyāyī is formulated. First of all, there is no critical edition of the Aṣṭādhyāyī of Pāṇini or other important earlier texts like the Mahābhāṣya of Patañjali—in which the Vārttika of Kātyāyana are also embedded—and later texts like the Kāśikāvṛtti of Jayāditya and Vāmana. All the pioneering editions in the last two centuries are vulgate editions without stemma. The issue was recently raised by Michael Witzel in an animated post to one of the electronic discussion groups, where he bemoaned the lack of efforts in preparing critical editions of basic Pāṇinian texts.³¹ Responding to this scholars like Johannes Bronkhorst downplayed the expectations that such critical editions might fulfil, for example, to shed some extra light on the exact date of Pāṇini or issues regarding the later additions of certain portions of Aṣṭādhyāyī.³² Madhav M. Deshpande, notes that “[T]his debate concerns a state of the Aṣṭādhyāyī that predates Kātyāyana and Patañjali, and no manuscript material takes us back to that period”. Therefore, in his opinion “crucial questions of historicity of various sections within the Aṣṭādhyāyī cannot be resolved with critical editions based on very late manuscripts.”³³

³¹ See the discussion group: Indo-Eurasian_research (http://groups.yahoo.com/group/Indo-Eurasian_research/) message 6303 posted on March 19, 2007 by witzel_michael (accessed on 12.01.2013).

³² Bronkhorst (2008a p. 475-484) supplies a detailed reply to Witzel on this issue.

³³ See the discussion group: Indo-Eurasian_research (http://groups.yahoo.com/group/Indo-Eurasian_research/) message 6311 posted on March 20, 2007 by deshbandem (accessed on 12.01.2013).

A joint project, however, is being carried out by École pratique des hautes études, Paris; Facolta di Studi Orientali, Università La Sapienza, Rome; and Bhandarkar Oriental Research Institute, Pune to bring out a critical edition of the Kāśikāvṛtti.³⁴ In this context, a notable publication is a critical edition together with translation of a section of Kāśikāvṛtti on *pratyāhāras*, which is an edited volume by Pascale Haag and Vincenzo Vergiani (2009). Despite these ongoing efforts, it would take some years till we have any fully fledged critical editions of some important early texts on Pāṇinian grammar. And even then, the question whether it may be helpful in resolving any question of significance, remains open.³⁵

The exact number of *sūtras* in the Aṣṭādhyāyī is not fixed. An approximate count is close to 4000. Moreover, as Hartmut Scharfe (2009 p. 33) notes: “[W]e have no independent assurance that the division of *sūtras* in our traditional text is always the one intended by Pāṇini”. Further, changes in the boundary of rules may lead to differences in the number and divisions of the *sūtras*. Sometimes a *sūtra*, which is traditionally given as one single rule, is divided into two for explaining the formation of certain words which otherwise are likely to be stamped as ungrammatical formations. This technique is called *yogavibhāga* and *sūtrabheda* (Abhyankar 1974 p. 318,432). Joshi and Roodbergen (1991 p. 20-23), for instance, split the rule *saṃbuddhau śākalyasyetāvanārṣe*³⁶ into two parts: *saṃbuddhau śākalyasya* and *itau anārṣe* and combine the two rules *uñah*³⁷ and *ūL*³⁸ into one.

There is, thus, no clarity about whether the verbal formulation of the grammatical system is without corruption. Even if one decides to begin with the grammatical corpus as laid down in modern editions of Aṣṭādhyāyī, and agrees on a *standard* version,³⁹ several major problems still persist which

³⁴ For more information, see: http://bori.ac.in/manuscript_department.html (accessed on 12.01.2015).

³⁵ Johannes Bronkhorst (2008a p. 482) remarks: “My expectation is that, even if all Pāṇinīyas were to mend their ways and spend their time making critical editions, and even if Witzel were to live to see the result, he might not find in (or through) these editions the answers he is looking for. To find these answers, other ways may have to be explored.”

³⁶ संबुद्धौ शाकल्यस्येतावनार्षे ॥ १.१.१६ ॥ ► according to Śākalya o(t) is termed *pragṛhya* if it is *saṃbuddhi* and when iti that is not Vedic (*ārṣa*) follows.

³⁷ उजः ॥ १.१.१७ ॥ ► according to Śākalya u(ñ) is termed *pragṛhya* when iti that is not Vedic (*ārṣa*) follows.

³⁸ ऊँ ॥ १.१.१८ ॥ ► according to Śākalya *anunāsika* ū replaces u(ñ) when iti that is not Vedic (*ārṣa*) follows and it is termed *pragṛhya*.

³⁹ There are different readings in modern editions as well and the task to prepare a *standard* version would be a time intensive exercise. Some efforts in this direction are made by Wiebke Petersen and Norbert Endres under the project “*Pratyāhāras* or features? A qualitative analysis of phonological descriptive techniques—a comparison of Pāṇini’s *pratyāhāras* and phonological features” (Personal communication). Project website: <http://panini.phil.hhu.de/?section=home> (accessed: 22.02.2013).

have more to do with the nature of the Aṣṭādhyāyī than the non-availability of its critical edition.

The rules of Aṣṭādhyāyī are not always stated in an explicit manner and require ample interpretation for their comprehension and application. The grammatical corpus needs to be supplemented with missing expressions, examples, counter examples etc. It is only then that it can be put to use. Mere recitation of the grammatical corpus (or in our case, inputting it in the computer) does not suffice. Not even resolving the *sandhis* and identifying the individual words. A user must exercise logical interpretations based on her or his knowledge of the grammatical system and the meta-linguistic conventions employed in the formulation of this system, in order to be able to apply it. Patañjali points out this in the following lines:

It is not that one derives linguistic expressions only through rules of grammar. Then how are the linguistic expressions derived? Through rules of grammar together with reasoned explanations (*vyākhyāna*). Now, if it is to be said that reasoned explanation is nothing but rules separated into constituent parts, then it is not correct, because it is not just dissected words such as *vṛddhiḥ āt aic* (of a rule like *vṛddhirādaic*⁴⁰). What, then, is reasoned explanation? It consists of examples, counter examples, completion of statements by filling the missing words. Reasoned explanation is all this combined together.⁴¹

It follows that one needs to provide mechanisms for incorporating the reasoned explanation (*vyākhyāna*) as well, if the formalization is to be put to application. This, however, is not exhaustively mentioned in the rules of grammar. In other words, the grammatical corpus of Aṣṭādhyāyī does not explicitly incorporate the understanding of its application. Therefore, for the purpose of application of grammar, mere rules of Aṣṭādhyāyī are not enough.

Another problem which makes the task of formalization of the Pāṇinian grammar difficult is the lack of consistent application of its meta-linguistic conventions. This point can be best presented through the detailed study of the principles of *anuvṛtti* (carrying over of earlier components to subsequent rules) by S. D. Joshi and Saroja Bhate. It shows that although the conventions for *anuvṛtti* hold good for a number of rules, there are several counter examples. To quote Joshi and Bhate (1984 p. 252):

The conventions of *anuvṛtti*, which are called rules in a loose sense, are valid in the majority of cases. We do not claim that they are valid in each and every case. [...] There are counterexamples which go against the conventions of *anuvṛtti*, yet the most salient fact about the assumption of these conventions is that they can not be given up, even in the face of apparent counterexamples. These contradictory examples do

⁴⁰ वृद्धिरादैच् ॥१.१.१॥ ► vṛddhi stands for āt and aic.

⁴¹ न हि सूत्रतः एव शब्दान्प्रतिपद्यन्ते । किं तर्हि । व्याख्यानतः च । ननु च तदेव सूत्रं विगृहीतं व्याख्यानं भवति । न केवलानि चर्चापदानि व्याख्यानं वृद्धिः आत् ऐच् इति । किं तर्हि । उदाहरणं प्रत्युदाहरणं वाक्याध्याहारः इति एतत्समुदितं व्याख्यानं भवति । (P. 122-124).

not disprove our conventions. Therefore we claim that the apparent counterexamples are clearly manifestations of later insertions.

The reason why Joshi and Bhate do not want to give up the conventions is that otherwise the functioning of the grammatical system would be impossible. At the same time the presence of counter-examples shows the impediments to the process of formalization. On the basis of the inconsistencies in the systematic use of the principles of *anuvṛtti*, they propose later insertions of rules in the Aṣṭādhyāyī. In particular, they point out that insertion of the Vedic rules and *nipātana* rules, the rules dealing with samāsa and taddhita formations leads to the irregularities in the application of the conventions of *anuvṛtti*.

The hypothesis that there must be successive additions is further corroborated by the presence of “conflicting and incompatible elements in different parts of the text”. In this regard, the following remark of Joshi and Bhate (1984 p. 253) is significant:

For instance, the same suffix is referred to differently. The instrumental suffix (ṭ)ā (4.1.2) is referred to as ā(ṅ) in 7.3.105 and 7.3.120. The accusative dual suffix au(ṭ) is referred to as au(ṅ) in 7.1.18. The Aṣṭādhyāyī (for instance, 7.2.9) refers to certain elements which are not prescribed in its basic layer. Instead of the genitive case, the nominative is used in the samāsa-section to indicate that A becomes B (i.e. *sthānyādeśabhāva*).

Even if one is ready to ignore or amend the irregularities in consistent application of the meta-level conventions, it does not suffice for a direct formalization of the corpus of Aṣṭādhyāyī. The reason for this lies in the very nature of some of these conventions. As an example, one may quote a few of the principles specified by Joshi and Bhate in case of *anuvṛtti*:

Sometimes to have a proper interpretation of the rule the locative case is to be changed into the nominative. (Convention no. 18c). Topics or sections need not necessarily be introduced by the section-heading rules. They can be understood on the basis of Pāṇini’s style of phrasing the rules. (Convention no. 16l). If the same section heading (*adhikāra*) occurs in different sections, it must have some reason to do so. (Convention no. 18k).⁴²

An examination of the above conventions makes it evident that although many of them are conducive for recognizing certain patterns in the formulation of the rules, they are not sufficient for a formal representation which a computer can understand and implement.

One further handicap is the language of the Aṣṭādhyāyī itself. The grammatical corpus is in Sanskrit. This means, the rules of the grammar are applied to the language of the grammar as well. Yet they are applied selectively and not in a consequent manner. Not all the rules that are applied for formation

⁴² For the list of these conventions, see (Joshi and Bhate 1984 p. 271-279).

of linguistic expressions are applied in the formulation of grammatical expressions. In the words of S. D. Joshi and J. A. F. Roodbergen (1991 p. 2):

Pāṇini's *sūtra*-language differs from ordinary, literary Sanskrit in this that the rules followed in this type of Sanskrit are not necessarily applied to the *sūtra*-language also. In literary Sanskrit a *samāhāradvā* ending in /-c/ takes the *samāsānta*-suffix (t)a(c) (5.4.106).⁴³ For instance, *vāktvācam* "the aggregate of voice and skin". But this rule is not applied to *ādaic*. An important reason for not applying Pāṇini's own rules to his own *sūtra*-language is the concern to keep the expression clear and unambiguous.

In other words, the rules of Aṣṭādhyāyī are applied to the language of Aṣṭādhyāyī only as long as clarity of the grammatical corpus is not undermined.

Another problem is the loss of meta-linguistic information, especially the accentuation and nasalization of the grammatical elements. According to the convention *svaritenādhikārah*⁴⁴ a circumflex accent (*svarita*) indicates the beginning of a domain (*adhikāra*). But as P. S. Subrahmanyam (1999 p. 182) notes:

The technical use of the circumflex accent was lost in later times along with the general loss of accent in the language. The author of *Kāśikā* (7th century A.D.) acknowledges this loss when he remarks: *pratiññāsvaritaḥ pāṇinīyaḥ* which amounts to saying "the circumflex accent is understood to be present only there where the Pāṇinian scholars think that it should be present".

Moreover, the question as to which vowel of an *adhikāra-sūtra* is accented is also not clear. According to Subrahmanyam (1999 p. 182):

Patañjali did not touch this problem. But the author of the *Kāśikā* and his commentators (i.e. the authors of the *Padamañjarī* and *Nyāsa*) think that all the vowels of an *adhikāra-sūtra* have the circumflex accent.

The same problem is faced once one wants to decide which sounds in a component are it-markers. The meta-rule *upadeśe'janunāsika it*⁴⁵ says that the nasal vowels of a component should be an it-marker. But the nasalization of vowels in the original instruction was also lost and *Kāśikāvṛtti* notes that "the Pāṇinīyas decide about the nasalization on the basis of established agreement on this".⁴⁶ Similarly, accents (*udātta*, *anudātta* and *svarita*) were also used to mark the verbal roots. In this case as well, the accentuation was lost and later grammarians had to make separate lists of roots carrying these accent markers.⁴⁷ Meta-rules also do not cover all the eventualities. The set of meta-rules specifying the sounds that are it-markers in the original

⁴³ द्वन्द्वाच्चुदषहान्तात्समाहारे ॥५.४.१०६॥ ► the suffix (t)a(c) is introduced after a dvandva compound ending in *cu*, *d*, *ṣ* or *h* provided it expresses collection.

⁴⁴ स्वरितेनाधिकारः ॥१.३.११॥ ► through *svarita* a domain is marked.

⁴⁵ उपदेशेऽजनुनासिक इत् ॥१.३.२॥ ► in grammatical instruction, an *ac* which is *anunāsika* is *it*.

⁴⁶ प्रतिज्ञाऽनुनासिक्याः पाणिनीयाः । (KV on 1.3.2).

⁴⁷ In the *Dhātupāṭha* the listing is sub-categorized under *udāttaḥ*, *udātta-itaḥ*, *anudātta-itaḥ*, *svarita-itaḥ* etc. See, for example, (Katre 1989 p. 1173-1200).

instruction do not include all the cases.⁴⁸ Kātyāyana notes this fact and suggests that the group *ir* added at the end of verbal bases in the Dhātupāṭha should be called it.⁴⁹ Patañjali gives *rudh(ir)* as an example, and clarifies that it is not enough to designate *i* and *r* one after another as markers, but the entire group *ir* should be deleted together.⁵⁰ It follows that one can not depend upon the meta-linguistic rules to decide the it-markers. An explicit annotation is necessary.

The arguments noted above confirm that the strong version of the formalization hypothesis is not tenable. In other words, it would not be possible to input the text of Aṣṭādhyāyī and get as output a formal representation which a computer program can understand or implement. This does not negate the fact that Aṣṭādhyāyī is itself an attempt to present the grammatical system in a formal manner. Still, it was meant for oral transmission and human application.⁵¹ It would be anachronistic to expect that it should also fit the requirements of machines developed some two-and-a-half millenia later. Moreover, any such claim would ignore the efforts of both later Pāṇinīyas and the works of modern researchers to amend, explain and bring consistency in the corpus through several conventions.

3.2.1 Possibility of formalization of the grammatical system

The problems faced while attempting to provide a formal representation of the corpus of Aṣṭādhyāyī in a direct manner lead one to look for other ways. Here the following differentiations need to be recognized. Firstly, it is necessary to differentiate between the system and the framework within which this system is articulated. Secondly, it is important to distinguish between the oral/verbal framework in which Aṣṭādhyāyī is formulated and a formal framework which a computer program expects. Finally, it is essential to recognize the features of an organizational framework in contradistinction to the characteristics of an applicational framework. As these distinctions

⁴⁸ The following rules specify it-markers: उपदेशेऽजनुनासिक इत् ॥१.३.२॥ ► in grammatical instruction, an *ac* which is *anunāsika* is *it*. हलन्त्यम् ॥१.३.३॥ ► *hal* coming at the end. न विभक्तौ तुस्माः ॥१.३.४॥ ► but not in *vibhakti*, the *tu*, *s* or *m*. आदिर्षिटुडवः ॥१.३.५॥ ► the initial *ñi*, *ṭu* and *ḍu* (of *dhātu*). षः प्रत्ययस्य ॥१.३.६॥ ► *ṣ* of *pratyaya*. चुटू ॥१.३.७॥ ► *cu* or *ṭu*. लशक्तद्धिते ॥१.३.८॥ ► *l*, *ś* and *ku* of non *taddhita*.

⁴⁹ V 4 on 1.3.7.

⁵⁰ See (Joshi and Roodbergen 1994 p. 14). Pāṇini refers to entities with *ir* as it-markers in इरितो वा ॥३.१.५७॥ ► After *irit dhātu* replace (c)[i] through a(ñ) optionally before *parasmaipada* substitutes of *l(u)(ñ)* expressing *kartṛ*.

⁵¹ For arguments that the Aṣṭādhyāyī was developed keeping only oral representational apparatus at hand see also (P. 155-163) in (Joshi and Roodbergen 1986 p. 40).

are central to my approach for providing a formal representation of the Aṣṭādhyāyī, I examine them briefly.

The first distinction—namely between the system and the framework—corresponds to the difference between the questions as to *what* is being told and *how* it is being expressed. In the present case the questions would be: what is the grammatical system and how is it presented in the grammatical corpus. The conventions through which a framework is regulated correspond to the meta-linguistic conventions of the Aṣṭādhyāyī.

As an example, consider the use of it-markers. In the corpus of Aṣṭādhyāyī Pāṇini invests specific grammatical characteristics to the constituent components by attaching marker sounds to them. The Pāṇinian term for these indicators is “it” (meaning: that which goes away or disappears) and the commentators use the term *anubandha* as well.⁵² The it-markers are not a part of the actual form of the components and are deleted unconditionally.⁵³ Their temporary character is noted by Patañjali who compares them with a crow perched on a house:

If someone asks now, “from among the two houses, which one is the house of Devadatta”, one can say “that on which the crow is perching”. And later, even if the identifying mark of that house is no more, once the crow flies away, the questioner recognizes the house.⁵⁴

The above remark of Patañjali points out that even after the it-markers are deleted, their identifying character is *retained* or *saved* by the user in her or his memory. Why must they then be deleted at all? Several problems may otherwise arise.

Firstly, marker sounds increase the number of phonemes in a component. This poses a difficulty where, for example, components with only one phoneme are allowed. Therefore, a meta-rule is formulated which says that while counting the number of phonemes, the markers should not be counted.⁵⁵

Secondly, markers may influence the final phoneme of a component. For example, the root *dai(p)* in fact ends with *ai* and not *p*. It is important to delete the marker sound *p* otherwise the long vowel substitute *ā* can

⁵² This term “was chosen for mute significatory letters by ancient grammarians probably on account of the analogy of *anubandha paśu*, tied down at sacrifices to the post and subsequently slaughtered” (Abhyankar 1986 p. 25). Cf., e.g., MB on 1.1.26.

⁵³ Their unconditional elision is provided by the rule: तस्य लोपः ॥१.३.९॥ ► its elision (takes place).

⁵⁴ कतरत् देवदत्तस्य गृहम् । अदः यत्र असौ काकः इति । उत्पतिते काके यदि अपि नष्टं तत्गृहं भवति अन्ततः तं उद्देशं जानाति ।(MB on 1.1.26). Translation with minor variations from (Subrahmanyam 1999 p. 92).

⁵⁵ नानुबन्धकृतम् अनेकाल्त्वम् (PŚ. 6).

not replace the final sound *ai* of the verbal root as prescribed by the rule *ādeca upadeśe'siti*⁵⁶. An extra meta-rule is stated to clarify this. It posits that while looking at components ending with *ec* the markers should be ignored.⁵⁷

Thirdly, markers also change the form of the components and may cause problem while deciding the similarity in form of two components. For example, the suffixes (k)a and a(ṅ) have the same form only if one ignores the markers. Therefore, an extra meta-rule is mentioned which says that while deciding the (dis)similarity of forms, markers should be ignored.⁵⁸

As is evident, these problems are because of the fact that the addition of markers changes the *form* of the components. Grammatical conditions, however, take the form of a component without markers into consideration and therefore, they must be unconditionally deleted immediately after their introduction in the derivational process. Their effect or function, however, must be *remembered*.

This takes us to the second distinction, namely between the oral and formal frameworks. The critical question is how to record or represent the grammatical information in a formal framework. The text of Aṣṭādhyāyī was recited and then later put in writing in more than one script. The step which is needed for representing oral text in a written corpus needs to be taken further to represent it in a formal framework. Continuing with the example, the marker sounds which in roman transliterations of the corpus are noted by capital letters⁵⁹, or bold face letters⁶⁰, or put into brackets⁶¹, can be recorded as an attribute attached with the corresponding component.

In the new framework, it-markers are represented as attributes attached to the form of corresponding components. Instead of mentioning it at the form level and thus mixing it with the form of the constituent components, they are placed at the content level together with other attributes.⁶² For example, the markers ṅ and k attached with the phonetic form of the suffixes a(ṅ) and (k)a are not stored at the form level but at the content level as attributes *ṅit* and *kit*.

Separating the two levels and maintaining this distinction in the repre-

⁵⁶ आदेच उपदेशेऽशिति ॥६.१.४५॥ ► *āt* replaces *ec* occurring at the end of verbal roots enunciated in the original recitation, except when a suffix with *ś* as it-marker follows.

⁵⁷ नानुबन्धकृतम् अनेजन्तत्वम् (PŚ. 7).

⁵⁸ नानुबन्धकृतम् असारूप्यम् (PŚ. 8). This decision is important for application of the suffixes (k)a as an exception to the suffix a(ṅ). See (Subrahmanyam 1999 p. 187).

⁵⁹ Sumitra M. Katre's edition of the Aṣṭādhyāyī (1989) uses this convention as also the volumes by S. D. Joshi and J. A. F. Roodbergen (1975) etc.

⁶⁰ George Cardona (1997) follows this in his edition.

⁶¹ This is used by P. S. Subrahmanyam (1999).

⁶² See section 2.2.1 for the form and content levels.

sentation of grammar has several advantages. The meta-rules mentioned in (PŚ. 6-8) are now redundant. The Pāṇinian rule *tasya loṣaḥ*⁶³ becomes redundant as well. This is because the marker attribute now being at the content level does not change the form of the component. Moreover, there are instances when a component with a marker loses its marker and is used without it. In other words, sometimes a particular marker is *removed* although originally it constitutes a part of the component. Thus by the rule *na ktvā seṭ*⁶⁴ the it-marker k is removed. This is easily executed by removing the corresponding attribute kit from the component.

Further, the new representation also makes it possible to take care of situations where the characteristic of having a marker is carried over to a component, although the marker sound is not expressly attached to it. For example, the rule *sārvadhātukamapit*⁶⁵ says that a *sārvadhātuka* suffix which is not pit (or not marked with p) is as if *ñit* (or marked with ñ). This amounts to attaching the attribute *ñit* to the appropriate component.⁶⁶

At this point one may ask whether I am suggesting a new jargon to note down the Aṣṭādhyāyī? This is not the case for the following reasons: firstly, it is not that I am only providing new transliteration conventions or merely inputting the corpus into the computer. Such efforts provide for editions which only human readers can read, interpret and apply. On the other hand, a formal representation renders the grammar in such a manner that an algorithm can read, interpret and apply it. Secondly, in order to achieve this, the applicational aspects of the grammar need to be incorporated with the organizational concerns. Aṣṭādhyāyī is formulated with the aim of optimal organization of the grammatical content and processes. Its application is left to the user. This is natural as the prime concern was to record the information and since the system is learnt by a student, its application is not explicitly stated.

However, it does not mean that Aṣṭādhyāyī is oblivious to application. The rules are specified in an operational set-up, where the derivational process is carried out. The rules of Tripādī, or last three sections of the final chapter of Aṣṭādhyāyī, have a clear procedural thrust as well.

It is much later in the 14th and 15th centuries CE that the task of reorganization of the Aṣṭādhyāyī with procedural application (*prakriyā*) as its main focus was taken up. The two most important texts which attempt this are the *Prakriyākaumudī* of Rāmacandra (late 14th-15th cent.) followed by

⁶³ तस्य लोषः ॥ १.२.९ ॥ ► its elision (takes place).

⁶⁴ न क्त्वा सेट् ॥ १.२.१८ ॥ ► (k)tvā together with i(t) loses kit.

⁶⁵ सार्वधातुकमपित् ॥ १.२.४ ॥ ► *sārvadhātuka* pratyaya, if not pit, are assigned *ñit*.

⁶⁶ In the Aṣṭādhyāyī the rules 1.2.1 to 1.2.26 provide for the assignment of it-attributes.

the Siddhāntakaumudī of Bhaṭṭoji Dīkṣita (late 16th-17th cent.). Both of them rearrange the *sūtras* of the Aṣṭādhyāyī for the purpose of derivation of forms.⁶⁷

The approaches of grammarians like Rāmacandra and Bhaṭṭoji Dīkṣita however are not without problems. Pāṇinian *sūtras* make sense only in their original context. A *sūtra* is not an independent unit and for the sake of application several inter-connected *sūtras* need to be taken into account. A reorganization disturbs the original contextual location of the *sūtras*. This takes us to the third distinction, namely, how some information is organized, i.e. the organizational framework and how it can be put to use, i.e. the applicational framework. A formal representation takes care of both aspects and records them in an explicit and non-ambiguous manner. This facilitates an implementation through a computer program. Keeping the above distinctions in mind, I propose the following hypothesis.

Weak version of the formalization hypothesis: The Aṣṭādhyāyī in its current formulation cannot be formalized directly but the grammatical structure which it encapsulates can be reformulated in a formal framework. The formalized reformulation can then serve as an input to a computer program that can interpret and implement it.

3.3 Specification of the formal framework

The discussions in the previous sections show that a formal framework needs to be developed in order to represent the content and processes of Aṣṭādhyāyī if one wants logical systems to interpret and apply it. An important feature of such a framework has to be a clear distinction and separation between the content of the grammar and the manner in which it is formulated. In other words, the framework should be independent of the content of grammar. It should only facilitate the representation of the structures and the system, without in any way influencing them.

There are two basic concepts upon which the present framework is based. The first concept is that any given *whole* can be analyzed into *components*. The second is the idea that any given entity can be described through a number of characterizing *attributes*.

While proposing the manner in which the grammatical elements—namely the components, attributes and meaning-expressions—should be represented

⁶⁷ Compared to the earlier attempt of Rāmacandra, Bhaṭṭoji Dīkṣita covers all rules of the Aṣṭādhyāyī and follows them consequently. See (Houben 2008 p. 563-574).

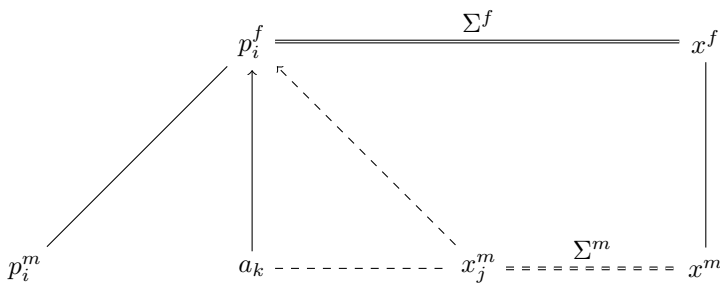
in the new framework, the way in which they are employed during the process of derivation is also taken into account. Similarly, the simulation of the derivational process takes into consideration the manner in which the grammatical elements are represented. The focus, therefore, is to integrate the organizational and applicational aspects of grammar.

As discussed in section 2.3 the derivational process demands a mechanism through which the changing forms and contents of the grammatical elements can be tracked in a convenient manner. Equally important is to keep track of the process of derivation, i.e. the previous stages attained during the process, and record them in a manner so that they can be consulted when deciding the future steps. Moreover, central to the Pāṇinian process is the interaction with the user. While specifying the formal framework, these general guidelines need to be taken into consideration.

The new framework which I put forward is designed to represent the categories and processes outlined in the previous chapter about the grammatical system. Three kinds of basic elements are proposed:

1. The constituent components p_i of the language.
2. The diverse attributes a_k which are either explicitly mentioned or sometimes implicitly used in the grammar.
3. The copious meaning-expressions x_j^m that encode a multitude of information which is external to the grammatical system.

While attributes and meaning-expressions work at the *content-level*, the *form* of a component is of utmost relevance for the derivation of the linguistic expressions. This is depicted in the following diagram.



There are two fundamental *processes* in this formulation.

1. Specification of the components. A component is characterized by associating the information from the content level. Thus, some meaning can be established through usage p_i^m , or some grammatical attribute a_k may be attached to it, or some meaning-expression x_j^m may be associated with

it. Further, some specific form—e.g. presence of some phonetic feature—characterizes a component as well.

2. Combination of the components. This involves introduction of new components in the derivational process.

3.3.1 Representing a component

The first task is to specify how the constituent components can be represented. For this purpose it is necessary to look into how these are formulated within the corpus of Aṣṭādhyāyī and whether the new representation adequately incorporates all the aspects of Pāṇinian formulations.

In section 2.2.1 we have seen that the constituent components p_i of a linguistic expression x consist of two levels, namely the form and the content levels. At the form level these are clearly made up of a sequence of sound units. An obvious manner to represent them would be a sequence or list. For example, the component bhū can be seen as a sequence of two sounds bh and ū which is represented as

[bh, ū]

Aṣṭādhyāyī contains “the first enunciation” (*upadeśa*) of the constituent components of Sanskrit.⁶⁸ Looking from the perspective of the form level one can speak of two distinct parts in the Aṣṭādhyāyī according to the two fundamental types of the constituent components—the basic sounds and secondly their sequences.

1. In the beginning of the Aṣṭādhyāyī the first part enumerates the fundamental sounds. These listings are called the Śivasūtras.
2. The rest of the Aṣṭādhyāyī enunciates components of the language which are a sequence of one or more basic sounds. These collections are called the Sūtrapāṭha, the Dhātupāṭha and the Gaṇapāṭha.

The listing of vowels in the Śivasūtras corresponds to a group of sounds and each one of them is further specified by additional attributes. For example, the sound a stands for several variations of this fundamental sound, namely short (hrasva) /a/, long (dīrgha) /a/ etc. If one represents a sound through a set, then the general sound /a/ would be represented by {a} and the long vowel /ā/ by {a, dīrgha}. In other words, adding more attributes to the set

⁶⁸ The word *upadeśa* refers to the original or first instruction of the grammatical components in the corpus before it enters and changes its form through the derivational process (*prakriyā*). For a detailed study on the concept of *upadeśa* in Sanskrit grammar see (Biswal 1996).

which represents a sound would result in further restricting and specifying a particular sound. Thus, {a, dīrgha, udātta, anunāsika} denotes a particular, more specific instance of the vowel /a/.

A sound can be represented through a set consisting of a *fundamental-sound* and a number of other characterizing *attributes*. It should be noted that within a set representing any sound, there can be *exactly one* fundamental-sound. In the following, I will call such a set a *sound-set*.

In the above example, the fundamental sound is /a/ and the characterizing attributes are dīrgha, udātta etc. The Śivasūtras provide the set of fundamental-sounds.

{a, i, u, r, l, e, o, ai, au, h, y, v, r, l, ñ, m, ñ, n, ñ, n, jh, bh, gh, dh, dh, j, b, g, ḍ, d, kh, ph, ch, ṭh, th, c, ṭ, t, k, p, ś, ṣ, s, h}

The form of any constituent component of a linguistic expression can now be represented as a sequence of sound-sets. Consider again the example of the component bhū. It can be represented as a sequence of two sound-sets:

$$bhū = [\{bh\}, \{u, dīrgha, udātta\}]$$

This representation allows one to include any number of attributes that may characterize the sounds. The attributes that are associated with a sound include phonetic and phonological features.⁶⁹ Thus, one can include other attributes e.g. the attribute *vowel* for which the Pāṇinian term is *ac*.

$$bhū = [\{bh\}, \{u, dīrgha, udātta, ac\}]$$

In this manner the information which is specific to a particular sound can be incorporated in the corresponding sound-set.

Some information, however, is shared by more than one sounds. For example, the attribute dhātu (verbal root) is valid for both sounds of this component. Accordingly, this attribute can be included in both sound-sets.

$$bhū = [\{bh, dhātu\}, \{u, ac, dīrgha, udātta, dhātu\}]$$

The sound-sets in a sequence share those attributes that are common to them. The presence of the attribute dhātu in both of these sets indicates that both of them in this sequence are part of dhātu or a verbal root.

In the above sequence of sound-sets, thus far, there is no reference to

⁶⁹ For an introduction to feature systems see (Clark, Yallop and Fletcher 2007 p. 372-390). James Stanton Bare (1980) has studied the system of features implicit in the Aṣṭādhyāyī. Interpretations of the Pāṇinīyas on the question of nearness (*antaratamya*) is discussed by Robert A. Hueckstedt (1995).

bhū, the original component which is being represented here. This information can be included by adding bhū to both the sound-sets that form a part of this component.

$$X = [\{bh, dhātu, bhū\}, \{u, ac, dīrgha, udātta, dhātu, bhū\}]$$

The linguistic component can now be summarised as X , and all the information can be gathered by the sound-sets that are contained in it.⁷⁰

The above example leads us to specify a general representation for the components.

$$X = [\{sd_1, a_1, a_2, \dots\}, \{sd_2, a_1, a_{17}, a_{43}, \dots\}, \{sd_i, a_k, \dots\}, \dots]$$

Here, X is a sequence of sets. Each set consists of *exactly one* key from the collection of fundamental sounds. Thus, the first set has sd_1 , the second sd_2 etc. Further, each sound-set may have a number of attributes. Thus the first sound-set has attributes a_1 and a_2 etc. It should be noted that the unique fundamental sounds in each sound-set represent the basic phonetic form of that unit, and the attributes comprehend the characteristics shared commonly among several sound-sets or which are individual to a particular sound-set. For example, the attribute a_1 is common to the first two sound-sets and a_2 is unique to the first sound-set.⁷¹

Within the new framework, each element, i.e. the constituent components, attributes or meaning-expressions are assigned a unique identifier or an ID. There is an ID

- for every sound, like /a/ (ID: a_0), /i/ (ID: i_0), /u/ (ID: u_0) etc.
- for the phonetic form of every component: a (ID: a_2), bhū (ID: bhU_a), (ś)a(p) (ID: zap_0), ti(p) (ID: tip_0) etc.
- for the attributes: vṛddhi (ID: vRddhi_0), dhātu (ID: dhAtu_0) etc.
- for meaning-expressions: *vartamāna* (ID: vartamAna_x), and
- for lexical content: *bālaka* (ID: bAlaka_x).

Assignment of an ID ensures a non-ambiguous identification of the elements of grammar. For example, the substitute suffix a (ID: a_2) is distinguished from the phoneme /a/ (ID: a_0) on the basis of their IDs. However, for the sake of readability, I will not display the underlying IDs.

⁷⁰ There are a number of other attributes that are needed to be included in order to have complete information about the component bhū, e.g., that it is part of the first group of verbal roots etc.

⁷¹ Thus far, only one constraint is mentioned—namely the presence of exactly one of the fundamental sounds in each of the sound-sets. In due course, other constraints will be mentioned that are imposed by the grammar to specify particular components.

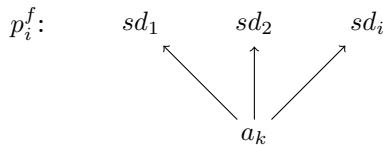
The components of any linguistic expression specified in the Pāṇinian corpus can be represented in terms of a sequence of sound-sets. Each of the sound-sets in such a sequence consists of *exactly one* ID from the set of fundamental-sounds. Further, it contains a number of other IDs corresponding to other attributes and meaning-expressions. Moreover, if this sound-set is part of some specific component, then this information is also incorporated by including the ID of that component with it. Within the new framework, such sequences of sound-sets are termed as language-components and can be defined as follows.

A language-component corresponds to any component of grammar and is a sequence of one or more sound-sets.

One of the features of the grammatical system specified in the previous chapter was the attachment of the attribute to some component.



In the above diagram, the attribute a_k is attached to the form of the components p_i^f . The present framework implements the attachment of the attributes in the following manner. If the form of some component has the sounds sd_1, sd_2 etc., then the attributes that are common to them are attached with each of the constituting sounds and are included in the respective sound-sets.



The lexical meaning p_i^m as well as the meaning-expressions x_j^m associated with some component are recorded in a similar manner. Thus, if a lexical meaning is connected with some component, then it is connected to each sound-set of that component.

In the following section I will provide examples to demonstrate that the components specified in the grammatical corpus of Aṣṭādhyāyī can be adequately represented in the new framework. Moreover, I will also point out how the Pāṇinian articulation differs from the new one, but at the same time all the information needed for grammatical processing is also included in my representation.

In the Pāṇinian formulation the fundamental sounds are represented directly. The new framework stores them through a sound-set with a unique identifier. Moreover, the phonetic attributes are stored explicitly through attribute IDs.

1. The sound /a/ in the Aṣṭādhyāyī will be equivalent to {a_0}. The long variation ā is equivalent to {a_0, dIrgḥa_0}. Pāṇini attaches the marker sound /t/ to restrict the time duration.⁷² Thus Pāṇinian āt is equivalent to {a_0, dIrgḥa_0} and at is equivalent to {a_0, hrasva_0}. One can also record the Pāṇinian attribute for short or long /a/, namely at or āt by including the IDs at_1 or At_2 to the corresponding sound-set. Thus {a_0, dIrgḥa_0, At_2} incorporates the information that the sound is called āt by Pāṇini. Inclusion of this information is redundant to some extent, but I include it to record the Pāṇinian term and to be able to correlate with the corpus of the Aṣṭādhyāyī.
2. Sounds like the nasal sound ṁ or the aspirated sound ḥ are not listed explicitly in the Śivasūtra but are referred to in the grammar by their names: anusvāra for ṁ and visarjanīya for ḥ. The new formulation includes the surface form of the sounds named through anusvāra or visarjanīya. Thus, the corresponding sound-sets are {M_1, anusvAra_0} and {H_1, visarjanIya_0}.
3. In the Pāṇinian formulation the constituent components can have additional indicatory sounds, namely the it-markers. These are attached either at the beginning or at the end of the base form of a component. For example, the suffix ti(p) consists of the sound p as marker at the end. Similarly, the infix (ś)a(p) has a marker sound ś in the beginning and p at the end. Sometimes more than one sound is added at the end or in the beginning, for example, yuj(ir) or (ḍu)krī(ñ).⁷³ In the new framework, the markers are included as an attribute. For example:

$$\begin{aligned} \text{ti(p)} &\cong \{[t, \text{pit}], [i, \text{hrasva, pit}]\} \\ (\acute{s})\text{a(p)} &\cong \{[a, \text{hrasva, } \acute{s}\text{it, pit}]\} \\ \text{yuj(ir)} &\cong \{[y, \text{irit}], [u, \text{hrasva, irit}], [j, \text{irit}]\} \end{aligned}$$

Thus, ti(p) is a component with the attribute pit i.e. “having p as it”. Similarly, (ś)a(p) is śit and pit and yuj(ir) is irit. The sign \cong refers to *partial equivalence* of the Pāṇinian component in the new formal framework.

⁷² तपरस्तत्कालस्य ॥ १.१.७० ॥ ► an ac followed by the marker t stands for sounds having the same time duration.

⁷³ Vidya Niwas Misra (1966) provides a list of these markers and the “functional load” associated with them. P. S. Subrahmanyam (1999 p. 92-108) uses the term indicators and notes them as well. All such marker sounds are represented by () brackets in our writing convention.

4. Within the Pāṇinian system, a sound can disappear, i.e. it can cease to be manifest at the audible level or the form level. The rule *adarśanaṃ lopaḥ*⁷⁴ states that the result of the attribute lopa is “non-appearance” (*adarśana*) of a sound. The same is the case with luk, ślu and lup which are attached only to suffixes.⁷⁵ The presence of attributes like lopa, luk, ślu and lup would imply that the corresponding sound-set is invisible or mute at the form level.
5. It is sometimes difficult to pronounce the constituent components as they consist only of consonant clusters without a vowel. In such cases, a vowel is added for facilitating pronunciation. This is necessary for an oral corpus. For example, the augmentation n[u](m) where the vowel u is just for the sake of pronunciation. In the printed edition, these are noted within square brackets []. Within the new framework, this information is redundant although it is stored for the sake of exact reproduction of the components in the original corpus.

Within the new framework, a component of any linguistic expression is represented in terms of a language-component, which is modeled as a sequence of sound-sets. These components can be seen as units of an inflected word.

For example, the word *paṭhati* has components paṭh(a), (ś)a(p), ti(p) etc. Although the derivational manuals like *Prakriyākaumudī* and *Siddhāntakaumudī* provide for derivation of individual words, the Pāṇinian system of Sanskrit grammar and especially its process of derivation functions not at the level of individual words, but at the level of an entire sentence. In the new framework, this aspect is taken care of by introducing the concept of a sentence and defining it as follows.

A sentence is a sequence of one or more language-components.

At this point it is important to mention that the above definition of a sentence refers to the collection of language-components within the new framework. It should not be confused with the linguistic definition of a sentence and should only be viewed as a sort of container in which the language-components can be stored in a sequential manner.

⁷⁴ अदर्शनं लोपः ॥ १.१.६० ॥ ► non-appearance is termed lopa.

⁷⁵ प्रत्ययस्य लुक्श्लुपः ॥ १.१.६१ ॥ ► non-appearance of a suffix is termed luk, ślu and lup.

3.3.2 Comprehending the dynamics of the derivational process

Thus far, a framework for representation of the Pāṇinian components has been introduced. The grammatical corpus not only provides the constituents of a linguistic expression but also the processes to combine them. This synthesis takes place in a number of steps. Within the traditional Pāṇinian framework this process is carried out by a person who takes decisions regarding the choices of introducing the new components. As mentioned earlier, the derivational process is guided by

1. the intention (*vivakṣā*) of the speaker, and
2. the constraints of the grammatical system, which involves
 - a. consistency of the derivational state
 - b. its saturation, and
 - c. its completion.

The new framework takes care of both the human and the grammatical aspects of the derivational processes.

Consider the example of derivation of the Sanskrit expression *bālakaḥ paṭhati* (a boy reads) worked out in the previous chapter. The first step is the introduction of the constituent components *bālaka* and *paṭh(a)*. Their representation in the new framework would be in terms of a sentence consisting of two language-components: if *X1* is the language-component for *bālaka* and *X2* for *paṭh(a)* then depending upon their mutual order, the sentence *S* is given by a sequence of *X1* and *X2*.

$$S = [X1, X2]$$

Further, each language-component is a sequence of sound-sets:

- $X1 = [\{b\}, \{a, dīrgha\}, \{l\}, \{a, hrasva\}, \{k\}, \{a, hrasva\}]$
- $X2 = [\{p\}, \{a, hrasva\}, \{th\}]$

For better readability, I note the language-components with their respective sound-sets in the following manner:

$$S = [X1, X2]$$

X1 ss1 {b, bālaka}

ss2 {a, dīrgha, bālaka}

ss3 {l, bālaka}

ss4 {a, hrasva, bālaka}

ss5 {k, bālaka}

ss6 {a, hrasva, bālaka}

X2 ss1 {p, paṭh(a), ait, udāttet}

ss2 {a, dīrgha, paṭh(a), ait, udāttet}

ss3 {ṭh, paṭh(a), ait, udāttet}

The sound-sets consist of a number of IDs. As mentioned earlier, an important constraint is that each sound-set must contain *exactly one* ID from among the set of fundamental sounds. For example, a sound-set like {a, i} is invalid. One can say that the collection of fundamental sounds forms a set of mutually exclusive IDs. Another example of a set of mutually exclusive IDs is the set {hrasva, dīrgha, pluta}. Within a particular sound-set the presence of an ID from this set excludes the possibility of inclusion of any of its other IDs. One can introduce the concept of consistency of a sound-set. One such condition of consistency would be that a sound-set should not have more than one ID from the set of mutually exclusive IDs.

There is a separation of the form and the content level. The language-component X2 represents paṭh(a). In the Pāṇinian formulation it has four sounds / p a ṭh (a)/ where the last one is only a marker, which is unconditionally deleted by the rule *tasya lopaḥ*⁷⁶. In the new formulation, the marker sound is not represented through an extra sound-set but is included as an attribute (ait) at the content level. Thus, only as many sound-sets are formed as there are sounds that contribute to the form of the linguistic expression.

Attachment of the attribute dhātu. The rule *bhūvādayo dhātavaḥ*⁷⁷ says that the components mentioned in the list beginning with bhū—i.e. the list of verbal roots or Dhātupāṭha—be assigned the attribute dhātu. In the new framework this assignment is carried out in the following manner: the sound-sets of the language-components are checked and if an ID which is also part of the set of the verbal roots is present, then that sound-set is attached the ID corresponding to the attribute dhātu. This can be formulated as follows: given a language-component Xi of some sentence S, if it contains one of the IDs belonging to the set {bhū, ..., paṭh(a), ...}, then attach the attribute dhātu.

S = [X1, X2]

X1 ...

X2 ss1 {p, paṭh(a), ait, udāttet, **dhātu**}

ss2 {a, dīrgha, paṭh(a), ait, udāttet, **dhātu**}

⁷⁶ तस्य लोपः ॥ १.३.९ ॥ ► its elision (takes place).

⁷⁷ भूवादयो धातवः ॥ १.३.१ ॥ ► components in the list beginning with bhū are dhātu.

ss3 {ṭh, paṭh(a), ait, udāttet, **dhātu**}

Here it should be noted, that in the new framework there is no need to introduce extra indices for different constituents as the IDs of components, attributes and meaning-expressions provide the requisite identification. So in this case, each sound-set of the language-components will be checked for whether it has some ID (in this case paṭh(a)) belonging to the set of the IDs of dhātus, and if this is the case then the attribute dhātu is added to that sound-set.

Similarly, attachment of other attributes e.g. prātipadika, kartṛ, pratyaya, vṛddhi etc. would bring about inclusion of the attribute IDs to the *appropriate* sound-sets. The selection of the appropriate sound-sets would depend upon examination of different conditions. Thus, attachment of an attribute results in addition of the IDs to one or more sound-sets.

Adding a new component l(a)(ṭ). Addition of a new component involves extension of the sentence by adding new language-components. The rule *var-tamāne laṭ*⁷⁸ introduces l(a)(ṭ) if present tense is intended. The derivational state looks as follows:

S = [X1, X2, X3]

X1 ...

X2 ss1 {p, paṭh(a), ait, udāttet, sakarmaka, dhātu}

ss2 {a, dīrgha, paṭh(a), ait, udāttet, sakarmaka, dhātu}

ss3 {ṭh, paṭh(a), ait, udāttet, sakarmaka, dhātu}

X3 ss1 {l, l(a)(ṭ), ṭit}

Here, a new language-component X3 is added after X2. For this, the indices of the language-component with dhātu need to be identified and the new component should then be introduced after it.

Replacement of l(a)(ṭ) by ti(p). Replacement of a component by another can now be represented as a combination of addition and attribute attachment. Thus, in order to replace l(a)(ṭ) by ti(p), the language-component X4 corresponding to ti(p) is added after the language-component X3, which corresponds to l(a)(ṭ). Further, a new attribute δ (denoting “replaced”) is attached to X3. The derivational state looks as follows:

S = [X1, X2, X3, X4]

X1 ...

⁷⁸ वर्तमाने लट् ॥ ३.२.१२३ ॥ ► to express present time, introduce l(a)(ṭ).

X2 ...

X3 ss1 {l, l(a)(t), tit, abhihita-kartṛ, δ }

X4 ss1 {t, ti(p), pit, prathama, ekavacana}

ss2 {i, hrasva, ti(p), pit, prathama, ekavacana}

It should be noted that the component l(a)(t) which has been replaced is not dropped from the sentence. Instead, a new attribute δ is attached to it. In this manner the information which it carries remains available for all further steps. In the traditional framework, owing to its oral and linear nature, the substituted component needs to be removed and in its place, the new replacement is placed.

The process of derivation progresses through a number of derivational states. A particular derivational state represents the effect of some grammatical operation on the current sentence. The grammatical operations are enjoined by the operational rules of grammar. Within the new framework, the operational rules of Aṣṭādhyāyī are modeled as statements (see next chapter).

A derivational state, therefore, stores the effect of a statement when applied on a sentence.

The above example indicates that from the perspective of the new framework, there are two basic types of operations which the operational rules enjoin:

1. Attachment of an attribute to some language-component or a sound-set.
2. Addition of a component to some language-component.

The first category corresponds to the process of saturation of a derivational state and the second one for its completion. A number of saturating statements are applied giving rise to a sequence of derivational states. A container is required to collect these derivational states. Such a sequence of derivational states is termed a *slice*. Thus, a slice is defined as follows.

Slice is a sequence of derivational states.

Attributes are attached as long as the level of saturation is not reached. Once no more attribute can be attached to a particular derivational state, then the current slice attains saturation. Then, depending upon the conditions, a new component can be added which prompts an introduction of a new slice. As long as new components can be introduced, the process remains incomplete. Again, a container is required to collect the slices. Once no more component can be introduced it attains completion.

The sequence of slices is collected in a *process-strip*. Thus, a *process-strip* is defined as follows.

Process-strip is a sequence of slices.

To sum up, the entire derivational process is modeled through a *process-strip* which is a sequence of slices. Whenever a new component is added, a new slice is introduced. Within a slice, there is a sequence of *derivational states*. Each *derivational state* corresponds to the attachment of some attribute. Finally, a *derivational state* stores the effect of some *operational statement* on a sentence. Further, a sentence is a sequence of *language-components* which in turn consists of a sequence of *sound-sets*.

Chapter 4

Representations

In the previous chapter I proposed a new framework and it is time now to represent the operational rules of Aṣṭādhyāyī in this new framework. In contrast to the *sūtras* of Aṣṭādhyāyī which are meant to be interpreted and applied by a learned student of grammar, the operational rules are formulated in terms of statements. In the following, I will take a representative selection of the operational rules of Aṣṭādhyāyī and render them in the new framework. The task to represent all the operational *sūtras* of Aṣṭādhyāyī would require more time and space than I have at my command at this moment. My claim, however, is that the operational rules of the entire Aṣṭādhyāyī can be represented in this manner.

4.1 Operational statements

4.1.1 Characterizing the components

Consider the very first *sūtra* of Aṣṭādhyāyī: *vṛddhirādaic*. It says that *āt* and *aic* are termed *vṛddhi*. Traditionally, it is a definition (*sañjñā-sūtra*) specifying the technical term *vṛddhi*. In the new framework it is seen as an operational statement that provides for attachment of the attribute *vṛddhi* to those sounds that contain the attribute *āt* or *aic*. It is specified in the following manner.

+a: vṛddhi

1. $X_m+a \parallel X_m[\bar{a}t, aic]$

वृद्धिरादैच् ॥ १.१.१ ॥ ► vṛddhi stands for $\bar{a}t$ and aic .

The above statement consists of three lines. The first one is the header where the variable +a represents the type of operation, namely attachment of an attribute. This is followed by a semi-colon : and then the value i.e. the ID of the attribute which is to be attached. The attribute whose attachment is ordained by this rule is vṛddhi. So the first line says that this statement provides for attachment of the attribute vṛddhi.

The second line specifies that the attribute a is meant to be attached to a sound-set X_m . This is noted as X_m+a . Here, X_m is the m-th sound-set of some language-component X. This is followed by two parallel vertical lines || which separate the actual listing of the conditions. The condition in this case is noted as $X_m[\bar{a}t, aic]$. It means that the sound-set X_m should have either the ID: $\bar{a}t$ or the ID: aic . The second line is numbered as 1. The need for numbering will become clear once we meet cases where more than one mutually exclusive condition may trigger the operation. The core of the statement consists of the first two lines. It can be read as follows: attach an attribute with value vṛddhi. The attachment is to a sound-set. This sound-set must contain either the ID: $\bar{a}t$ or the ID: aic .

The third line which follows is useful for quick reference. It contains the original Pāṇinian *sūtra* which corresponds to the above statement together with an English translation.

The above statement is formulated within the formal framework introduced in the previous chapter. Thus, it provides for attachment of an attribute to a sound-set. Moreover, the condition which needs to be fulfilled is also specified in terms of the presence of certain IDs in that sound-set.

Two separate layers can be identified in the above statement. Firstly, its *structure* or *form* and secondly its *values* or *content*. The structure part is noted in the typewriter font. It is:

+a:—
 $X_m+a \parallel X_m[—]$

The content or values are the attributes vṛddhi, $\bar{a}t$ and aic . Thus, there is a clear separation between the form or structure of a statement on the one hand and its content or values on the other hand.

There are certain conventions employed in the new formulation. Some

of them correspond to the logical or structural properties of the statement and others are my choices for displaying them in print form. I collect and list these conventions in the grey boxes.

1. X stands for a language-component.
2. X_m stands for the m -th sound-set of some language-component X .
3. $+a$ stands for the type of operation, namely attachment of an attribute.
4. X_{m+a} says that the attribute a is attached to the sound-set X_m .
5. $X_m[ID1, ID2]$ represents the condition that the sound-set X_m should have either $ID1$ or $ID2$.

The next *sūtra* of Aṣṭādhyāyī is similar. In the grammatical corpus it is stated as *adeṅguṇaḥ*. It says that *at* and *eṅ* sounds are called *guṇa*. Its representation in terms of the new framework will be as follows:

+a: guṇa

1. $X_{m+a} \parallel X_m[at, eṅ]$

अदेङ्गुणः ॥१.१.२॥ ► *at* and *eṅ* are *guṇa*.

This statement can be read as follows: attach an attribute with value *guṇa*. The attachment is to a sound-set. This sound-set must contain either the ID: *at* or the ID: *eṅ*.

Now, if one compares the structure of this statement with the previous one, it becomes clear at once that both the statements have the following identical structure. Only the values of the IDs have changed.

+a: —
 $X_{m+a} \parallel X_m[—]$

The identical structure of the above statements tells us that from operational point of view, they are identical. In other words, both the statements can be implemented through a single operational function. In fact, the operation of attachment of an attribute to some component is a fundamental operation in the Pāṇinian system. Within the Pāṇinian corpus this operation is formulated and presented through different techniques. We list them in the following sections.

4.1.1.1 Explicit enumeration

A straight forward way to form a set is to enumerate its members explicitly. Pāṇini uses this technique extensively, where the components are listed together. As an example, consider the statement :

+a: niṣṭhā

1. $X+a \parallel X[(k)ta, (k)tavat(u)]$

कृत्तवत् निष्ठा ॥ १.१.२६ ॥ ► (k)ta and (k)tavat(u) are niṣṭhā.

Here the components (k)ta and (k)tavat(u) which form the set called niṣṭhā are mentioned explicitly. Again, the structure of this statement, namely

+a:—
 $X+a \parallel X[—]$

is similar to the structure of statements attaching the attributes vṛddhi or guṇa, namely:

+a:—
 $X_m+a \parallel X_m[—]$

There is, however, one important difference. This statement provides for attachment of the attribute niṣṭhā to the entire language-component, while the previous statements provide for attachment of attributes like vṛddhi or guṇa to a particular sound-set. Further, there are significant differences in the nature of values. While all the statements provide for attachment of some attribute, the conditions in the first two statements are formulated in terms of other attributes (like āt or aic), while in the case of the third statement, it is based on other fundamental components, namely (k)ta and (k)tavat(u). One can, at this point, speak of attributes that are attached to a sound-set or phonological features and those that are associated with a language-component or morphological characteristics.

Another example of attribute attachment to a language-component is:

+a: gha

1. $X+a \parallel X[tara(p), tama(p)]$

तरस्रमपौ घः ॥ १.१.२२ ॥ ► tara(p) and tama(p) are gha.

Apart from explicit enumeration of members of a set within the main

corpus, namely the Sūtrapāṭha of Aṣṭādhyāyī, there are extra lists or sets of components collected in the two large appendices—the Dhātupāṭha and the Gaṇapāṭha. Within these large lists there are several sub-groups as well. One common way of naming such a list is by mentioning its first element like bhū and then forming the compound *beginning with bhū* or bhvādi. The corresponding statement would be:

+a: bhvādi

1. X+a || X[bhū, paṭh(a), ..., ji, pū(n), dhe(t), ...]

The form of this statement is identical to the one for attachment of the attribute niṣṭhā. It says that whenever a language-component has any of the IDs for bhū, paṭh(a) etc., then the attribute bhvādi should be attached to it. All such sets that are specified in the grammatical corpus by explicit enumeration of their elements can be formulated within the new framework in a similar manner. For example, the statement for the next large set of verbal roots would be:

+a: adādi

1. X+a || X[ad(a), han(a), dviṣ(a), duh(a), dih(a), ...]

In both cases above, i.e. attachment of bhvādi or adādi, there is no mention of any *sūtra* of Aṣṭādhyāyī. This is because these statements do not correspond to any *sūtra* which is uttered explicitly in the grammatical corpus. Thus, the new formulation includes information that is specified in the grammatical corpus even if it is not mentioned in an explicit *sūtra* of the Aṣṭādhyāyī.

4.1.1.2 Formation of sigla (*pratyāhāra*)

The fundamental sounds are listed in what are called the Śivasūtras.¹ Compared to the listings of the sounds in Prātiśākhya, there are two novel features in Pāṇini's enumeration. Firstly, there is a new sequencing of the sounds and secondly, there are deleming markers placed between them. Following is the list of sounds in the Śivasūtras :

a i u ṛ ḷ k| e o ṛ ai au c| h y v r ṭ| l ṇ| ñ m ṇ ṅ n m| jh bh ṅ| gh ḍh
dh ṣ| j b g ḍ d ś| kh ph ch ṭh th c ṭ t v| k p y| ś ṣ s r| h l|

¹ The terms used for the list of units or phonemes are *akṣarasamāmnāya* (in Mahābhāṣya) and Śivasūtra or Maheśvarasūtra in later tradition. Nandikeśvara mentions in his Kāśikā that the lord Śiva sounded his drum (*dhakkā*) fourteen times revealing the fourteen *sūtras* to Pāṇini. Hence the name Śivasūtra or Maheśvarasūtra. See (Cardona 1997 p. 83). Important studies on the Śivasūtras include (Breloer 1935 p. 133-191), (Cardona 1969 p. 3-48), (Staal 1962 p. 1-10), (Kiparsky 1991a) and (Petersen 2004 p. 471-489 and 2009 p. 79-98).

This sequence is not according to the usual order found in the Prātiśākhya on the basis of place of articulation.² Moreover, certain consonants are inserted as delimiters. They are placed at particular positions indicating the final sound of a *sūtra*.³ Using delimiters, the groups of sounds can be named in a generic manner. For this, one begins from the initial non-delimiter sound and continues upto the final delimiter sound. All the non-delimiter sounds contained in this sequence belong to that group.⁴

The purpose of the special ordering as well as use of delimiters is to form groups and sub-sequences required for the functioning of grammar.⁵ Kātyāyana notes that “the teaching of the sounds is to provide a special ordering for the sake of functionality of rules”.⁶ Tradition uses the term *pratyāhāra* (lit. bringing together) for the groups formed in this manner using a marker sound as the end delimiter.⁷

Pāṇini performs the task of forming groups in a generic manner. Instead of enumerating individual groups, he provides a rule-based method for forming and naming them. The reason for this way of forming groups is brevity (*lāghava*).⁸ The technique employed for forming sets of sounds is employed for formation of groups of components as well. Examples include:

² Compare (RVPr. 1.1-14). B. Breloer (1929 p. 114-135) provides a detailed study of the organization of phonemes on the basis of place of articulation in comparison to the different ordering in the Śivasūtras. See also (Cardona 1965 p. 225-237).

³ These are also called it-markers. Usually the it-markers are attached to some component and this attachment is regulated by a set of meta-linguistic rules. In case of Śivasūtras however, they are placed as delimiters. In order to distinguish them from the it-markers appearing in the components, we write them in italics and not in ()-brackets. Cardona (1969 p. 12) employs a more appropriate formulation here than Joshi and Roodbergen (1986 p. 188), “placed” instead of “added”, as the it-markers are not attached to the previous sounds, they are just placed after them. They do not, for example, make the sound u in a i u ṅ | a ṅ it sound. P. S. Subrahmanyam (1999 p. 88-91) recognizes this in his notation of *pratyāhāras*.

⁴ This is specified by the rule: आदिरन्त्येन सहेता ॥ १.१.७१ ॥ ► an initial element together with the final it sound includes intervening elements.

⁵ The process which is at work here is to form sub-sequences. Given a sequence, a number of sub-sequences can be formed, not all of which are relevant for grammar specifications. Given a finite alphabet Σ^n upto 2^n sub-sequences, $\binom{n}{k}$ of length k can be generated, which need not all be different. The contiguous sub-sequences which can be generated are $\frac{n(n-1)}{2}$ in number. Pāṇini, however, employs 41 sub-sequences. There is a 42nd *pratyāhāra*, namely *ca*, mentioned by Kātyāyana in his Vārttika 3 under the rule नादिन्याक्रोशे पुत्रस्य ॥ ८.४.४८ ॥. See (Subrahmanyam 1999 p. 90).

⁶ वृत्तिसमवायार्थ उपदेशः (PV. 19). See (Joshi and Roodbergen 1986 p. 186-187).

⁷ Pāṇini does not use this term. See (Abhyankar 1986 p. 266-167).

⁸ Devasthali (1967) is a detailed study on the *anubandhas* of Pāṇini. The structure of the Śivasūtras and its explanation on the basis of brevity is worked out by Staal (1962 p. 1-10), Cardona (1969 p. 3-48), Kiparsky (1991a) and Petersen (2009 p. 79-98). Wiebke Petersen (2008) has proved using mathematical methods that the listing of units and the choice of positioning of markers are optimal.

$tiñ = [ti(p) \text{ tas } jhi \text{ si}(p) \text{ thas } tha \text{ mi}(p) \text{ vas } mas \text{ ta } \text{ātām} \text{ jha } thās \text{ āthām} \text{ dhvam}$
 $i(t) \text{ vahi } mahi(ñ)].$

$tañ = [ta \text{ ātām} \text{ jha } thās \text{ āthām} \text{ dhvam } i(t) \text{ vahi } mahi(ñ)]$

$sup = [s(u) \text{ au } (j)as \text{ am } au(t) \text{ (ś)as } (t)ā \text{ bhyām} \text{ bhis } (ñ)e \text{ bhyām} \text{ bhyas } (ñ)as(i)$
 $bhyām \text{ bhyas } (ñ)as \text{ os } \text{ām} \text{ ni } os \text{ su}(p)]$

$sut = [s(u), \text{ au}, (j)as, \text{ am}, \text{ au}(t)]$

Here the non-marker portion of the first component is taken as the initial part, and the last marker sound of the final component of the sequence as the final part.

Pratyāhāras are names of groups and sub-sequences. This indicates that they belong to the category of attributes. Forming groups of sounds and naming them implies that each element of that group is assigned an attribute, namely the name of that group. Thus the sounds a, i, u are assigned the attribute *añ*, since they form the group {a, i, u}. The rule *tasya lopah*⁹ which is applied for elision of the marker sounds does not apply in case of *markers* appearing in these names. It is for this reason that in my notation the marker sounds originally coming from a component, but now forming the name of a group, are not placed within brackets but are denoted by *italics*.

In formal representation, *pratyāhāras* and other names for groups of components are attributes that are attached to the respective sounds or components.

+a: ac

1. $X_{m+a} \parallel X_m[a, i, u, \text{ṛ}, \text{ḷ}, e, o, ai, au]$

अ इ उ ण् ॥१॥ ► a, i, u, (ñ). ऋ लृ क् ॥२॥ ► ṛ, ḷ, (k). ए ओ ङ् ॥३॥ ► e, o, (ñ). ऐ औ च् ॥४॥ ► ai, au, (c). आदिरन्त्येन सहेता ॥१.१.७१॥ ► an initial element together with the final it sound includes intervening elements.

+a: tiñ

1. $X+a \parallel X[ti(p), \text{ tas } jhi \text{ si}(p), \text{ thas } tha, \text{ mi}(p), \text{ vas } mas, \text{ ta } \text{ātām}, \text{ jha } thās, \text{ āthām}, \text{ dhvam}, i(t), \text{ vahi } mahi(ñ)]$

आदिरन्त्येन सहेता ॥१.१.७१॥ ► an initial element together with the final it sound includes intervening elements. तिस्रिङ्गसिप्थस्थमिष्वस्मस्तातांझथासाथांघ्वमिङ्गिमहिङ् ॥३.४.७८॥ ► $ti(p) \text{ tas } jhi \text{ si}(p) \text{ thas } tha \text{ mi}(p) \text{ vas } mas \text{ ta } \text{ātām} \text{ jha } thās \text{ āthām} \text{ dhvam } i(t) \text{ vahi } mahi(ñ)$ are the substitutes of the cover term *l* or a *lakāra*.

Sometimes the order of the sounds within a *pratyāhāra* is important.

⁹ तस्य लोपः ॥१.३.९॥ ► its elision (takes place).

Consider, for example, the rule *eco'yavāyāvah*¹⁰ where the order of sounds [e, o, ai, au] in *ec* must necessarily be maintained to correspond with the components [ay, av, āy, āv]. Sometimes, however, the order of sounds within a *pratyāhāra* is of no importance. For example, in the rule *vṛddhirādic*¹¹ the order of [ai, au] in *aic* does not need to be fixed.

The attributes *at*, *āt*, *īt*, *ūt* etc. characterize the respective length of the vowel. For example:

+a: āt

1. $X_m+a \parallel X_m[a][dīrgha]$

तपरस्तत्कालस्य ॥१.१.७०॥ ► an *ac* followed by the marker *t* stands for sounds having the same time duration.

The condition clause $X_m[a][hrasva]$ specifies that both the IDs, namely the one corresponding to *a* and the one to *hrasva*, must be present in the sound-set X_m . The following convention is used here:

6. $X_m[—][—]$ says that both the variables within the two square bracket must be simultaneously present within the sound-set X_m .

4.1.1.3 Use of attributes

Within the grammatical corpus, the names of smaller sets are used to constitute a larger one. For example, the rule *tiñ śit sārvaadhātukam* specifies that the elements of the sets *tiñ* and *śit* together form the set called *sārvaadhātuka*.¹² Now, as mentioned above, *tiñ* is a set of 18 suffixes and *śit* also is the name of the group of components having *ś* as it-marker. This can be seen as set union where a larger set is formed on the basis of the union of two or more smaller sets.

+a: sārvaadhātuka

1. $X+a \parallel X[tiñ, śit]$

¹⁰ एचोऽयवायावः ॥६.१.७८॥ ► *ec* = [e o ai au] are replaced by ay av āy āv respectively when *ac* follows.

¹¹ वृद्धिरादिच् ॥१.१.१॥ ► *vṛddhi* stands for *āt* and *aic*.

¹² Because of the *anuvṛtti* of *dhātoḥ* 3.1.91, strictly speaking, these suffixes are termed *sārvaadhātuka* once they are introduced after a verbal root. But for the sake of simplicity, I am not taking this into consideration here.

तिङ् शित् सार्वधातुकम् ॥ ३.४.११३ ॥ ► *tiñ* and those having *ś* as it marker are called *sārvadhātuka*.

The above representation says that if a language-component has either *tiñ* or *śit* as its attribute, then the attribute *sārvadhātuka* should be attached to it.¹³

4.1.1.4 The it-markers

The it-markers attached to a component give rise to a set which is named as *the set of components with a particular it marker*. For example, *śit* components are those that have *ś* as it, like *(ś)a(p)*, *(c)āna(ś)* etc. These are represented by including the attribute *śit* in the corresponding sound-sets. Thus, the new representation of the component *(ś)a(p)* would be:

$$X = \{[a, (ś)a(p), śit, pit, \dots]\}$$

The primary list of the components records this information about the it-markers by including attributes like *śit* or *pit* in the new representation of the component *(ś)a(p)*.

Sometimes components not originally marked with it-markers are assigned markers through specific rules. At other times, the characteristics of having a marker is removed in certain instances. We will address such dynamic assignment of it-markers later in sections 4.1.1.12 and 4.1.1.13.

4.1.1.5 Groups based on some identifying sound

Sometimes the set of components is specified by an indicatory sound which is shared by all the elements of the group. For example, in the rule *dādḥāghvoadāp* the intended components are the ones having the form *dā* or *dhā* with the exception of *dā(p)* and *dai(p)* i.e. *(ḍu)dā(ñ)*, *(ḍu)dhā(ñ)*, *dā(ñ)*, *de(ñ)*, *do*, *dhe(ṭ)*. Here the formulation of *sūtra* is not explicit and additional explanations are required to interpret it. There is no precise specification of the elements of the group. The intended components need to be enumerated in an explicit manner as follows:

+a: **ghu**

1. X+a || X[(ḍu)dā(ñ), (ḍu)dhā(ñ), dā(ñ), de(ñ), do, dhe(ṭ)]

¹³ In fact, here the condition that it must be a suffix or pratyaya which is introduced after a dhātu or verbal root is not mentioned. This is for the sake of simplicity and explanation.

दाघाघ्वदाप् ॥१.१.२०॥ ► (ḍu)dā(ñ), (ḍu)dhā(ñ), dā(ñ), de(ñ), do and dhe(ṭ) are ghu.

Thus, the instances where an explicit enumeration is sought, but not specified in clear terms are determined by exhaustive listing of the components meant to be listed.

4.1.1.6 Specification of the boundaries

A usual method employed in the Sūtrapāṭha of the Aṣṭādhyāyī is to specify the boundary within which the components that are mentioned are assigned some attribute. For example, the rule *prāgrīśvarānnipātaḥ*¹⁴ states that the components enumerated prior to *rīśvare* in the rule *adhirīśvare*¹⁵ are termed nipāta. Sometimes the boundary is not mentioned explicitly, but is indicated by the domain or *adhikāra*. For example, the attribute *pratyaya* is assigned on the basis of mentioning the domain in the rule *pratyayah*¹⁶, which says that the components introduced hereafter till the end of the fifth chapter are called *pratyaya*. It should be noted that there is no separate enumeration in this case, but the components are specified together with the conditions under which they are introduced and the associated operations that need to be carried out. Explicit statements are formulated in the new framework to attach such attributes.

+a: pratyaya

1. X+a || X[sa(n), (k)ya(c), kāmya(c), ..., l(a)(ṭ), ..., ti(p), ..., (ś)a(p), ...]

प्रत्ययः ॥३.१.१॥ ► pratyaya are components introduced subsequently.

4.1.1.7 Formation of groups of attributes

Pāṇini not only forms sets of components, he also defines the collections of attributes. For example, the attribute *kāraka* is the name of the group of attributes {kartṛ, karman, karaṇa, sampradāna, apādāna, adhikaraṇa}.

+a: kāraka

1. X+a || X[kartṛ, karman, karaṇa, sampradāna, apādāna, adhikaraṇa]

¹⁴ प्राग्ग्रीश्वरान्निपातः ॥१.४.५६॥ ► Before the rule 97 the units introduced are assigned nipāta.

¹⁵ अधिरीश्वरे ॥१.४.९७॥ ► adhi is assigned karmapravacanīya when it conveys lordship.

¹⁶ प्रत्ययः ॥३.१.१॥ ► pratyaya are components introduced subsequently.

It should be noted that although the semantic definition of *kāraka* is not considered here, as the *Aṣṭādhyāyī* does not provide it in an explicit manner through some meaning-expression, the term *kāraka* stands for the group of attributes mentioned above.

4.1.1.8 Specifying the complementary set

Pāṇini also specifies a set by defining it as complementary to some other sets. For example, in the rule *arthavadadhāturapratyayaḥ prātipadikam*¹⁷ the set of nominal stems (*prātipadika*) is defined as consisting of components that are meaningful (*arthavat*) and do not belong to the sets of verbal roots (*dhātu*) and suffixes (*pratyaya*).¹⁸

+a: *prātipadika*

1. $X+a \parallel XM[\textit{arthavat}] \wedge XNOT[\textit{dhātu}] \wedge XNOT[\textit{pratyaya}]$

अर्थवदधातुरप्रत्ययः प्रातिपदिकम् ॥ १.२.४५ ॥ ► a meaningful component which is not a *dhātu* and not a *pratyaya* is *prātipadika*.

All the above methods are different ways to specify the formulation of sets or sequences of components. The attribute in these cases is the name of the group and is attached to each element that belongs to this group.

Theoretically, it is possible to specify them in an explicit and precise manner by enumerating the set. Although, in the oral framework of Pāṇini there are some interesting and elegant ways to formulate formation of a group, it is neither precise nor explicit in all cases. In the new representation, all such instances are specified explicitly.

4.1.1.9 Naming the distribution of components

Apart from group formations, attributes also signify a specific distribution of the components in the derivational process. Consider the attachment of the attribute *saṃyoga*. It is stated as follows:

¹⁷ अर्थवदधातुरप्रत्ययः प्रातिपदिकम् ॥ १.२.४५ ॥ ► a meaningful component which is not a *dhātu* and not a *pratyaya* is *prātipadika*.

¹⁸ For the sake of simplicity we leave here other conditions mentioned in the rule *कृत्तद्धितसमासाश्च* ॥ १.२.४६ ॥ ► components ending with *kṛt* or *taddhita* or those which are *saṃāsa* as well.

+a: saṃyoga

1. $X_m X_{n+a} \parallel X_m[ha/] \wedge X_n[ha/] \wedge IDX[n=m+1]$

हल्लोऽनन्तराः संयोगः ॥ १.१.७ ॥ ► *ha/* in contiguous sequence are saṃyoga.

Here, $X_m X_{n+a}$ specifies that the attribute *a* should be attached to both the sound-sets, namely to X_m as well as X_n . Thus saṃyoga is attached to both the sound-sets. The condition $X_m[ha/]$ states that the sound-set X_m must contain the attribute *ha/*. Similarly, $X_n[ha/]$ says that the sound-set X_n must also contain *ha/*. Moreover, $IDX[n=m+1]$ specifies that the indices *m* and *n* are contiguous. The logical \wedge sign specifies that the three conditions are related with the logical AND operator. The new conventions that are needed here are:

7. $X_m X_n$ are consecutive sound-sets (phonemes). The order is FIXED.
8. $IDX[n=m+1]$ says index *n* is equal to the index *m+1* i.e. *n* is the next contiguous index following *m*.

It should be noted that this type of convention is a personal choice of the editor to note down the eventualities. However, from the point of view of complexity of grammar formulation, *the concept of consecutive sounds* is important.

The form or structure of this statement can be specified as follows:

+a:—
 $X_m X_{n+a} \parallel X_m[—] \wedge X_n[—] \wedge IDX[—]$

The form of this statement is different from forms of the previous statements. This implies that the attachment of an attribute like saṃyoga needs a very different operational function than attachment of attributes like *vṛddhi*, *guṇa* etc. In this case, it is the nature of the adjacent sound-set which would decide whether the two adjacent sound-sets can be assigned the attribute saṃyoga or not. This aspect of positional distribution requires the use of $IDX[—]$ clause. Here, however, there are no numbers as indices, but only the constraint that the sound-sets X_m and X_n need to be adjacent.

Another example is the rule *aco'ntyādi ṭi*. It assigns the attribute *ṭi* to that part of the sound sequence which begins with the last vowel. Thus, the attribute *ṭi* is attached dependent upon the distribution of the language-component. It is specified by the statement:

+a: ṭi

1. $X_b e+a \parallel X_b[ac] \wedge X_m \text{NOT}[ac] \wedge IDX[m=(b+1 \text{ TO } e)]$

अचोऽन्त्यादि टि ॥ १.१.६४ ॥ ► that part which begins with last ac is \dot{t} i.

This states that the attribute \dot{t} i is attached to a sub-sequence of the language-component X, namely from the sound-set X_b upto the sound-set X_e. Moreover, X_e is also the final sound-set of that language-component. Further, the initial sound-set of the sub-sequence should have the attribute ac and the subsequent sound-sets should not contain the attribute ac. In the representation above, the suffixes be denote a range from X_b upto X_e. Thus, X_{be} represents the sub-sequence [X_b, X_{b+1}, ..., X_e]. Further, X_m where the index IDX ranges from m = b+1 TO e says that the sub-sequence [X_{b+1}, ..., X_e] of sound-sets is meant. Here, the necessity of checking a particular range of sound-sets indicates that the assignment of the attribute sometimes depends upon the distribution of various components in a derivational state. The following new conventions are introduced:

9. X_{be} represents the range of sound-sets from X_b till X_e where X_e is the final sound-set of that language-component. X_m is a variable sound-set which ranges from the index b+1 till e.
10. X_mNOT denotes that the sound-set X_m should NOT contain any of the IDs mentioned in this condition clause.

The next attribute upadhā is attached to the penultimate sound-set of a language-component.

+a: upadhā

1. Xu+a || Xu[a/] ∧ Xe[a/] ∧ IDX[u=e-1]

अलोऽन्त्यात्पूर्व उपधा ॥ १.१.६५ ॥ ► an a/ which is penultimate is upadhā.

The subscript e in X_e indicates that it is the final sound-set of the language-component. Moreover, the condition under the IDX clause provides that Xu is the penultimate sound-set. As a convention:

11. Xu is reserved for the penultimate (upadhā) sound-set.

As the next example, consider the definition of laghu and guru. In the Aṣṭādhyāyī it is specified through the following three *sūtras* : *hrasvaṃ laghu, saṃyoge guru, dīrghaṃ ca*. Together, they say that a hrasva sound (short vowel) is laghu (light), except when saṃyoga (consonant cluster) follows, then it is guru (heavy), as also a dīrgha sound (long vowel).

From the perspective of the new formal framework, there are two instances of attachment of attributes, namely laghu and guru. Accordingly, there are two different statements. Consider first the attachment of the attribute laghu :

+a: laghu

1. $X_m+a \parallel X_m[\text{hrasva}] \wedge X_n\text{NOT}[\text{saṃyoga}]$

ह्रस्वं लघु ॥१.४.१०॥ ► hrasva is laghu. संयोगे गुरु ॥१.४.११॥ ► when saṃyoga follows, it is guru.

Both the *sūtras* 1.4.10 and 1.4.11 mentioned above need to be taken into account. It has to be made explicitly clear that a hrasva sound is only then laghu when saṃyoga does not follow. This information is implicit in the oral formulation. $X_n\text{NOT}[\text{saṃyoga}]$ notes this explicitly. It says that the next following sound-set X_n should not contain the ID saṃyoga. The two conditions are joined by the logical AND operator, which is represented in this book through the \wedge sign. Moreover, a convention is used here, namely that X_m and X_n represent two consecutive sound-sets.

The statement for attachment of guru is as follows:

+a: guru

1. $X_m+a \parallel X_m[\text{hrasva}] \wedge X_n[\text{saṃyoga}]$
2. $X_m+a \parallel X_m[\text{dīrgha}]$

ह्रस्वं लघु ॥१.४.१०॥ ► hrasva is laghu. संयोगे गुरु ॥१.४.११॥ ► when saṃyoga follows, it is guru. दीर्घं च ॥१.४.१२॥ ► dīrgha as well.

The conditions are now listed in two separate lines. They represent two disjoint cases when the attribute guru is to be attached. The first is when the sound is hrasva (short vowel) and is followed by saṃyoga or consonant cluster, and the second case is when the sound itself is dīrgha (long vowel). The individual lines contain the conditions that can be combined together with the logical OR operator. The reason for keeping them separate is not just for the sake of readability, but to distinguish between disjoint sets of conditions. This, as will be made clear, is important for noting the complexity of the conditions in an explicit manner (see section 4.2.2).

4.1.1.10 Associating attributes with meaning-expressions

The cases mentioned thus far are all dependent in one way or another upon the grammatical information stated in the grammatical corpus. The next big category of attribute assignment is through semantic characterization. This is usually achieved by associating the attributes with a meaning-expression.

The following statement associates the expression *mukhanāsikāvācana* (an utterance spoken simultaneously through mouth and nose) with the term *anunāsika*. In this case, the meaning of the expression *mukhanāsikāvācana* must be clear to the person using the grammar. In the present case, the user needs to be consulted in order to ascertain whether the sound represented by the sound-set X_m is *mukhanāsikāvācana* or not.

+a: anunāsika

1. $X_{m+a} \parallel X_m[a, i, r, l, u, e, ai, o, au, y, l, v] \wedge X_m M[mukhanāsikāvācana]$

मुखनासिकावचनोऽनुनासिकः ॥१.१.८॥ ► from mouth and nose uttered sound is *anunāsika*.

The new representation indicates the semantic character of the conditions by using the letter M in conditional clauses. Thus the M part in $X_m M$ says that the sound-set X_m takes the attribute *anunāsika*, if it fulfils the semantic condition judged through the user that the sound is uttered simultaneously through the oral and the nasal cavities (*mukhanāsikāvācana*). The first condition, namely, the sound-set X_m must contain one of the above mentioned sounds like a, i, r etc. is not mentioned explicitly in the original *sūtra*, but is included on the basis of the explanations of it. The following convention is used here:

12. $X_m M$ is a meaning-condition associated with X_m . It implies that the user needs to be consulted in order to ascertain the admissibility of the meaning-expression.

The next statement is of different kind. Instead of attaching an attribute it provides an answer to a particular kind of relation between two sound-sets. The relation to be tested is with respect to the attribute a.

?a: savarṇa

1. $X_p X_q ?a \parallel X_p X_q M[tulyāsyaprayatna] \wedge X_p[ac] \wedge X_q[ac]$
2. $X_p X_q ?a \parallel X_p X_q M[tulyāsyaprayatna] \wedge X_p[hal] \wedge X_q[hal]$

तुल्यास्यप्रयत्नं सवर्णम् ॥१.१.९॥ ► sounds with same articulatory effort are *savarṇa*.
नाञ्जलौ ॥१.१.१०॥ ► but not *ac* together with *hal*.

The *sūtra*: *tulyāsyaprayatnaṃ savarṇam* says that two sounds are *savarṇa* when they have the same place of articulation (*tulyāsyaprayatna*). Further, *nājjhalau* clarifies that a vowel (*ac*) and a consonant (*hal*) can not be *savarṇa*, even if they have same place of articulation. This implies that either both of them are *ac* or both of them are *hal* sounds. This is specified by the condition clauses $X_p[ac] \wedge X_q[ac]$ which posit that both the sound-sets must contain

ac. The other option that both of them should contain *hal* is comprehended in the next part of the statement. It must be noted that for the application of this statement, one needs to know what the places of articulation are and which sounds are articulated from which part of the mouth. Thus, the meaning-expression needs further explanations. The conventions used in this case are recorded as follows:

13. $X_p X_q$ are two sound-sets (not necessarily contiguous).
14. $X_p X_q ? a$ represents the question whether the sound-sets X_p and X_q are related with respect to the attribute a . The answer would be a boolean TRUE or FALSE.

4.1.1.11 Attributes based on intention

There is another category of attributes that are defined through a meaning-expression where the intention (*vivakṣā*) of the speaker plays a central role. For example, attachment of the attribute *karṭṛ* to any specific component shall depend upon the intention of the speaker, whether she or he intends to express *karṭṛ* through that component or not. Thus, although the rule *svatantraḥ karṭṭā* defines *karṭṛ* to be that *kāra* which is independent of others (*svatantra*), yet its attachment to a particular component depends upon the intention of the speaker.

+a: karṭṛ

1. $X+a \parallel XM[svatantra]$

स्वतन्त्रः कर्त्ता ॥ १.४.५४ ॥ ► that *kāra* which is *svatantra* or independent of others is *karṭṛ*.

Here M in the condition clause $XM[—]$ specifies that the intention of the speaker through the meaning-expression should be taken into consideration.

4.1.1.12 Extending an attribute

Sometimes, certain attributes are extended to components under particular circumstances. Tradition calls these provisions *atideśa* (extention) rules.

+a: ñit

1. $X_i+a \parallel X_h[gā(ñ), kuṭādi] \wedge X_i[pratyaya] \wedge X_i \text{NOT}[\ñit][ñit]$

गाङ्कटादिभ्योऽञ्जिण्डित् ॥ १.२.१ ॥ ► after gā(ñ) and kuṭādi those pratyaya not having ñit or ṇit are assigned ñit.

Here the positions of X_h and X_i have to be in this order. That means a suffix in the language-component X_i which has neither ñit or ṇit and which comes after the component X_h, having either the component gā(ñ) or the attribute kuṭādi, is attached the attribute ñit.

4.1.1.13 Removing an attribute

There are cases when an attribute is removed under certain conditions. Consider the following example:

-a: kit

1. X_i-a || X_g[śī(ñ), ṣvid(ā), (ñi)mid(ā), (ñi)kṣvid(ā), (ñi)dhṛṣ(ā)] ∧ X_h[i(t)] ∧ X_i[niṣṭhā]

निष्ठाशीङ्घिदिमिदिद्विदिधृषः ॥ १.२.१९ ॥ ► niṣṭhā preceded by i(t) and coming after śī(ñ), ṣvid(ā), (ñi)mid(ā), (ñi)kṣvid(ā), (ñi)dhṛṣ(ā) loses kit.

Here, instead of adding, an attribute is removed. This is denoted by -a: which says that the attribute following after this should be removed from the appropriate language-components. The condition clauses specify the constellation. There are three language-components, X_g, X_h and X_i in this order. The language-component from which the attribute kit should be removed is X_i, specified by the attribute niṣṭhā, which according to the rule *ktaktavatū niṣṭhā*¹⁹ stands for the components (k)ta and (k)tavat(u). Further, this will happen only when it is preceded by X_h having the augment i(t). The whole complex should follow X_g with one of the components listed in it.

4.1.2 Combining the components

The process of derivation involves the combination of the constituent components with one another. This leads to a change at the form level. A change at the form level can happen when a component is introduced. Further, changes can also be induced by replacement of a component, augmentation or when some of the phonetic features are changed, for example, when a short vowel becomes long.

¹⁹ क्तवत् निष्ठा ॥ १.२.२६ ॥ ► (k)ta and (k)tavat(u) are niṣṭhā.

4.1.2.1 Introducing a component

Within the formal framework, the combination of components is modeled through a fundamental operation, namely introduction of a new language-component. Consider the introduction of the suffix $l(a)(t)$.

<y: $l(a)(t)$

1. $X < y \parallel X[dhātu] \wedge yM[vartamāna]$

वर्तमाने लट् ॥ ३.२.१२३ ॥ ► to express present time, introduce $l(a)(t)$.

The basic structure of the above representation is similar to the attachment of the attributes. In the first line, however, the sign < represents the operation of introducing a component whose ID is mentioned after the colon. The second line contains the conditions. Its first part specifies the position where the new component should be placed. In this case, after a language-component X which has the attribute $dhātu$. Moreover, the second condition specifies that the component $l(a)(t)$ is introduced when *vartamāna* (present tense) is intended to be expressed. The user needs to be consulted for this decision. Hence it is indicated by the variable M in yM .

4.1.2.2 Replacement of a component

Combination of components also involves replacement of a component by another component. Consider the replacement of the *lakāra*, say $l(a)(t)$, by the suffix $ti(p)$. Within the new framework, it can be represented as follows:

<<y: $ti(p)$

1. $Xi << y \parallel Xh[dhātu][parasmaipada] \wedge Xi[lakāra] \wedge yM[prathama]-[ekavacana]$

लस्य ॥ ३.४.७७ ॥ ► in place of suffixes with cover term l namely $l(a)(t)$, $l(i)(t)$, $l(u)(t)$, $l(r)(t)$, $l(e)(t)$, $l(o)(t)$, $l(a)(ñ)$, $l(i)(ñ)$, $l(u)(ñ)$ and $l(r)(ñ)$ the suffixes coming in the next rule are substituted. तिप्तद्ध्वसिप्थस्थमिप्थस्मस्तातांझथासाथांध्वमिद्ध्विहिहिद्ध्व ॥ ३.४.७८ ॥ ► $ti(p)$ tas jhi $si(p)$ thas tha $mi(p)$ vas mas ta $ātām$ jha $thās$ $āthām$ $dhwam$ $i(t)$ $vahi$ $mahi(ñ)$ are the substitutes of the cover term l or a *lakāra*.

The basic structure of this representation is again similar to the previous one. The operation, however, is replacement and the symbol <<y indicates this. The first line specifies the replacement operator followed by the replacement variable y , which after the colon is followed by the ID of the component. The

second line lists the conditions. The locus of the component which is to be replaced is specified in the first part before the vertical lines. In this case the component which is to be replaced is X_i , and the replacement is noted by the variable y . Finally, the conditions are stated. The two components X_h and X_i should be in this order. This is recorded in the meta-linguistic conventions of the new formulation.

15. X_h and X_i indicate that the two components have to be in this order.

Although a different operation, replacement is not a fundamentally new operation, and in the formal representation it can be considered as a combination of introduction of a component and attachment of an attribute. Those sound-sets that get replaced are attached the attribute δ . The new component is placed after the replaced component. In the example above it would imply that the replaced language-component X_i gets the attribute δ and the new component $t_i(p)$ is placed after it.

4.1.2.3 Augmentation

Another operation is augmentation (*āgama*) by phoneme-sequences that do not carry any specific meaning for the sake of attaining the desired surface form. For example, the rule *ārdhadhātukasyeḍvalādeḥ* provides the initial increment $i(t)$ which is inserted at the head of an *ārdhadhātuka* suffix that begins with a *val* phoneme.²⁰

<y: $i(t)$

1. $y > X_i \parallel X_h[dhātu][aṅga] \wedge X_i[ārdhadhātuka] \wedge X_{ib}[val]$

आर्धधातुकस्येड्वल्लादेः ॥७.२.३५॥ ► the increment $i(t)$ is inserted at the head of an *ārdhadhātuka* affix which begins with a *val* phoneme.

This is modeled in the same manner as the introduction of a new component.

16. $y > X_i$ indicates that the attachment here is an augmentation.

17. In X_{ib} the suffix b represents the initial sound-set.

²⁰ The *val* phonemes are {v, r, l, ñ, m, ṅ, n, jh, bh, gh, dh, dh, j, b, g, d, d, kh, ph, ch, ṭh, th, c, ṭ, t, k, p, ś, ṣ, s, h}

4.1.2.4 Change of phonetic form

The form of a language-component may also change if the phonetic attributes are modified. For example, the short (hrasva) vowel may be changed to a long (dīrgha) vowel. Similarly, the form level is also modified by the non-appearance of a sound. A component, or parts thereof, may undergo elision (lopa) i.e. non-appearance at the form level. For example, the rule *lopo vyorvali*²¹ says that the sounds v or y are elided if a suffix beginning with a val sound follows.

4.2 Structure of a statement

Statements follow a fixed syntax. The structure of a general statement can be specified as follows:

```
Operation-Declaration:ID
1. Placement-Distribution || ANDeD Conditions
2. Placement-Distribution || ANDeD Conditions
3. Placement-Distribution || ANDeD Conditions
...
```

There are two main divisions. The first line, depicted in bold-case letters, declares the unique operation and the other lines appear as a numbered list in the printed version, and specify an independent set of conditions that need to be met in order to carry out the above operation.

Within the operational declaration there are two parts separated by a colon. The first part specifies the nature or type of the operation. For example, attachment of an attribute is noted by +a. The other part specifies the ID of the component, either the ID of the attribute which is to be attached, or of the constituent component which is to be added.

After the header line, there may follow one or more lines, appearing as a numbered list in the printed version, that provide conditions for the above operation. Each of the numbered condition lines constitutes a particular case, when the operation specified in the header line can be carried out. The conditions specified in different lines are mutually disjoint.

A condition line itself consist of two parts separated by || two vertical lines. The first part consists of the structure or distribution of the derivational state and the location where the attachments or additions are to be made. The

²¹ लोपो व्योर्वलि ॥ ६.१.६६ ॥ ► lopa replaces v or y if a suffix beginning with val follows.

second part records the conditions in terms of IDs that need to be present. The conditions within a line are ANDed. The different lines combined together with the logical OR cover all the cases when a particular operation is to be performed.

The locus where a component should be added or an attribute should be attached is specified clearly using the exact variables for the corresponding sound-sets or language-components. The meta-rules in the Aṣṭādhyāyī regarding the locus of grammatical changes are taken care of by explicitly specifying the relative positions of the variables and use of sub-scripts.

The formal framework facilitates new methods and parameters to look into the complexity of grammar and its application for the derivational process. This can be formulated in terms of the types of operations, conditional complexity of a statement as well as inter-dependence and inter-relation between the statements.

4.2.1 Types of operations

The processes specified in the Aṣṭādhyāyī can be summarised in terms of the new framework in the following types:

- +a: Attachment of an attribute. An attribute can be attached to one or more sound-sets, or to an entire language-component.
- a: Removing an attribute. There are instances when some attribute is taken out or removed from a sound-set or a language-component. For example in certain cases the attribute kit is removed (see the rules 1.2.7-1.2.26).
- ?a: Checking the consistency. As a consistency check, statements are modeled as a question whose answer can either be TRUE or FALSE.
- <y: Addition of a component.

These are the four fundamental types of operations. Other operations like replacement or elision can be implemented as a combination of the above fundamental operations. Thus,

- <<y: Replacement or substitution can be implemented as a combination of (i) attachment of a new attribute δ to that part which is replaced and (ii) addition of the component which is the replacement at the appropriate place. Elision or lopa would involve attaching the attribute lopa to the concerned parts in the language-component. The sound-sets containing the attribute lopa would then not be expressed at the form level which is the audible layer.

4.2.2 Complexity of a statement

Consider again the representation of attachment of the attribute *vṛddhi* in the new framework.

+a: vṛddhi

1. $X_{m+a} \parallel X_m[\bar{a}t, aic]$

वृद्धिरादैच् ॥१.१.१॥ ► *vṛddhi* stands for *āt* and *aic*.

Leaving aside the grammatical elements, the structural form of this statement would be as follows:

+a:—
 $X_{m+a} \parallel X_m[—]$

We can term this as the *signature* of the above statement. The signature of a statement comprehends primarily the structural complexity of the conditions that need to be satisfied in order to apply that statement.

Looking at the following representation of attachment of the attribute *guṇa*, it is apparent that it has the same signature as the previous statement for attachment of the attribute *vṛddhi*.

+a: guṇa

1. $X_{m+a} \parallel X_m[at, eñ]$

अदेङ्गुणः ॥१.१.२॥ ► *at* and *eñ* are *guṇa*.

Let us now consider attachment of the attribute *laghu*.

+a: laghu

1. $X_{m+a} \parallel X_m[hrasva] \wedge X_nNOT[samyoga]$

ह्रस्वं लघु ॥१.४.१०॥ ► *hrasva* is *laghu*. संयोगे गुरु ॥१.४.११॥ ► when *samyoga* follows, it is *guru*.

The corresponding signature would be as follows.

+a:—
 $X_{m+a} \parallel X_m[—] \wedge X_nNOT[—]$

Similarly, for attachment of the attribute *guru*, we have the following statement.

+a: guru

1. $X_{m+a} \parallel X_m[\text{hrasva}] \wedge X_n[\text{samyoga}]$
2. $X_{m+a} \parallel X_m[\text{dīrgha}]$

ह्रस्वं लघु ॥१.४.१०॥ ► hrasva is laghu. संयोगे गुरु ॥१.४.११॥ ► when samyoga follows, it is guru. दीर्घं च ॥१.४.१२॥ ► dīrgha as well.

The signature for attachment of guru is as follows.

+a:—
 $X_{m+a} \parallel X_m[—] \wedge X_n[—]$
 $X_{m+a} \parallel X_m[—]$

If one now compares the statements for attachment of vṛddhi or guṇa with that of laghu or guru, one notices that the operation is the same in all cases, namely attachment of an attribute to some sound-set. What is different is the nature and complexity of the conditions. There are three different types of structures for the conditions:

$X_{m+a} \parallel X_m[—]$
 $X_{m+a} \parallel X_m[—] \wedge X_n\text{NOT}[—]$
 $X_{m+a} \parallel X_m[—] \wedge X_n[—]$

Attachment of many of the attributes discussed before have signatures similar to the one mentioned above. In contrast, attachment of attributes like samyoga, upadhā or ṭi that depend on the distribution of adjacent sounds have very different signatures.

+a: samyoga

1. $X_m X_{n+a} \parallel X_m[\text{hal}] \wedge X_n[\text{hal}] \wedge \text{IDX}[n=m+1]$

ह्रलोऽनन्तराः संयोगः ॥१.१.७॥ ► hal in contiguous sequence are samyoga.

The corresponding signature would be:

+a:—
 $X_m X_{n+a} \parallel X_m[—] \wedge X_n[—] \wedge \text{IDX}[—]$

The conditions for grammatical operations can be summarized as a combination of the following basic types:

The intention (*vivakṣā*) of the speaker plays a central role in the introduction of

many components. It is formulated through meaning-expressions and is usually stated in the locative case. For example, consider the introduction of the suffix $l(a)(t)$. This is introduced if the speaker intends to express present tense.

<y: l(a)(t)

1. $X <_y \parallel X[\text{dhātu}] \wedge yM[\text{vartamāna}]$

वर्तमाने लट् ॥ ३.२.१२३ ॥ ► to express present time, introduce $l(a)(t)$.

The signature of this statement is as follows:

<y: —
 $X <_y \parallel X[—] \wedge yM[—]$

The second part of the condition clause $yM[—]$ denotes that the intention of the speaker communicated through the meaning-expression is required for the application of the statement.

The presence (or sometimes absence) of attributes often provides conditions for the grammatical operations. For example, in the above statement for introducing the suffix $l(a)(t)$, the first part of the condition clause $X[—]$ requires the presence of the attribute *dhātu* (verbal root) in the component to which the suffix can be added. The presence of specific components quite frequently conditions the grammatical operations.

<y: sa(n)

1. $X <_y \parallel X[\text{gup}(a), \text{tij}(a), \text{kit}(a)]$

गुप्तिज्किञ्च्यः सन् ॥ ३.१.५ ॥ ► after $\text{gup}(a)$, $\text{tij}(a)$, $\text{kit}(a)$ introduce $\text{sa}(n)$.

Here, the condition clause $X[—]$ is similar in structure to the previous example. The only difference lies in the type of elements that populate this condition. In the previous case it was the *attribute*, and in the present example these are the specific *components*, namely, $\text{gup}(a)$, $\text{tij}(a)$ and $\text{kit}(a)$.

A specific constellation or distribution of the components also provides a condition for certain operations. Take the example of attachment of the attribute $\ddot{t}i$.

+a: $\ddot{t}i$

1. $Xb e + a \parallel Xb[ac] \wedge X m \text{NOT}[ac] \wedge \text{IDX}[m = (b+1 \text{ TO } e)]$

अचोऽन्त्यादि टि ॥ १.१.६४ ॥ ► that part which begins with last ac is $\ddot{t}i$.

The condition has to take into account a specific distribution of sound-sets, namely that part which begins with the last vowel and extends to the end of the language-component. Similarly, the rule *yasmātpratyayavidhistadādi pratyaye'ṅgam*²² specifies aṅga to be that portion after which a suffix is prescribed and when the suffix follows. Such instances of attribute attachment, therefore, require *looking* at the current derivational state and, based upon how the different components are placed there, deciding whether a particular attribute should be attached or not. These are dynamic attributes.

Specific grammatical processes may also form a condition. In certain cases not only a particular derivational state needs to be looked at, but the progress or change from one state to the next is important.

For example, the rule *igyaṇaḥ samprasāraṇam*²³ terms samprasāraṇa to the *ik* sounds, i.e. [i, u, r, l] which come in place of *yaṇ* i.e. [y, v, r, l]. In order to represent conditions of this kind, we need to look into the *history* of a particular component. This is facilitated by looking into the derivational states and slices that store the derivational process.

Mutual relations of certain grammatical processes or changes also set conditions. For example, the rules listed in the last three sections of the eighth chapter of the Aṣṭādhyāyī. By the rule *pūrvatrāsiddham*²⁴ the rules listed later are in effect suspended with respect to previous rules. Consider the following situation in which a derivational state st_1 advances to the next state st_2 after application of a rule A.

$$A(st_1) \rightarrow st_2$$

If there is another rule B which is related to the rule A in such a manner that A is suspended (*asiddha*) with respect to B, it implies that B can not operate on the state st_2 even if the conditions for the application are fulfilled. Since A is suspended with respect to B, it means that for rule B it is as if A was not applied at all. So only the state st_1 is visible to B. The implementation of the statements from this section would again require access to the derivational *history* by accessing the earlier derivational states.

²² यस्मात्प्रत्ययविधिस्तदादि प्रत्ययेऽङ्गम् ॥ १.४.१३ ॥ ► that part which enjoins a pratyaya based operation, before that the sequence is aṅga.

²³ इग्यणः सम्प्रसारणम् ॥ १.१.४५ ॥ ► *ik* replacing *yaṇ* is samprasāraṇa.

²⁴ पूर्वत्रासिद्धम् ॥ ८.२.१ ॥ ► that which follows from here onwards is treated as if suspended in view of what precedes.

4.3 Relation between statements

The signature of a statement specifies its structural form and comprehends the form of the conditional clauses. The manner in which any two components of grammar are related can now be formulated in terms of the categories of the new framework. Consider again the statement for attachment of the attribute $vṛddhi$:

+a: $vṛddhi$

1. $X_{m+a} \parallel X_m[\bar{a}t, aic]$

वृद्धिरादैच् ॥ १.१.१ ॥ ► $vṛddhi$ stands for $\bar{a}t$ and aic .

The condition clause requires the presence of the attribute $\bar{a}t$ or aic . This implies a hierarchy between $vṛddhi$ on the one hand and $\bar{a}t$ or aic on the other hand. This would also imply that the statement that attaches the attribute $vṛddhi$ will only then be relevant once the statements for attachment of $\bar{a}t$ or aic have been applied. The statements that attach the attributes $\bar{a}t$ and aic to some sound-set are as follows:

+a: $\bar{a}t$

1. $X_{m+a} \parallel X_m[a][dīrgha]$

तपरस्तत्कालस्य ॥ १.१.७० ॥ ► an ac followed by the marker t stands for sounds having the same time duration.

+a: aic

1. $X_{m+a} \parallel X_m[ai, au]$

ऐ औ च् ॥ ४ ॥ ► $ai, au, (c)$. आदिरन्त्येन सहेता ॥ १.१.७१ ॥ ► an initial element together with the final it sound includes intervening elements.

One can say that the statement for attachment of the attribute $vṛddhi$ presupposes the application of the statements for the attachment of the attributes $\bar{a}t$ or aic .

Given two statements: $St-A$ and $St-B$, $St-B$ is dependent on $St-A$ if the application of $St-B$ requires the results of the application of $St-A$.

The *dependency graph* for the components and attributes of Aṣṭādhyāyī would provide an order in which the statements will have to be applied. The problem of inter-dependence or circularity, i.e. $St-B$ is dependent on $St-A$ and $St-A$ is dependent on $St-B$, can be resolved by introducing a new statement, let us say, $St-B1$ with a different ID.

The examples from the operational rules of Aṣṭādhyāyī which I have represented in the new framework show that a formal representation of Aṣṭādhyāyī is possible. It is also possible to reduce the role of an individual for applicational purposes, although grammar functions integrally together with the speaker or user. The new representation also aims to clearly separate the form or structural complexity of the grammar from the inter-relatedness of its components. It provides a mechanism to look into these relations at the more granular level of individual components, in contrast to attempts to organize them within the categories of semantics, syntax, morphology and phonology. The present framework allows a flexible integration of information which is either not explicitly mentioned in the grammatical corpus or which can be included from other sources, e.g. the inclusion of the suggestions of Kātyāyana in his Vārttika or the Uṇādi-sūtras.

Chapter 5

Implementation

A computer implementation of the content and processes of Aṣṭādhyāyī can be aptly compared with the automatic machines for buying railway tickets. When interacting with such a machine, a traveller needs to supply specific information about the destination, route, date and time, number of travellers, class etc. Depending upon the interactive design of the program this information is required by the machine as and when it needs to take some decision for which the wish and consent of the traveller is necessary. On its part, the machine can put constraints upon the choices depending upon the current situation, e.g. whether an option to take a particular route or train on a particular day is possible or not. Finally, the desired ticket is printed.

In a similar manner, a computer program that implements the derivation of a linguistic expression requires input by the user regarding her or his intention (*vivakṣā*). This information is necessary to make appropriate decisions so that finally the desired expression is formed. The program, on its part, puts constraints upon the possible choices that are admissible.

In the following, I provide an overview of the computer implementation of the Pāṇinian system. As discussed in the previous chapters, my rendering of the grammar is through its formal representation. Therefore, the basis of the present computerisation is the formalisation discussed in chapters 3 and 4.

The programming codes specified here aim to show that my model of the Aṣṭādhyāyī facilitates its computer implementation. Because of the constraints of the space, there is no attempt to furnish the details which run into several thousand lines of program codes. Moreover, thus far there is no user interface for the system. This is because it does not directly contribute to the research questions and would simultaneously require considerable time which is beyond the scope of the present work. The programs are written in the Python Programming Language which runs on Windows, Linux/Unix, Mac

OS X and is available under open source license.¹ Needless to mention, there can be different ways of implementation depending upon the overall design of the program, the nature of the programming environment as well as the choices of the implementational platform.

There are two main aspects of the computer implementation which I will deal with in this chapter. Firstly, how to implement the grammatical content on a computer, and secondly, how to simulate the processes of derivation. The former has to deal with the data-structures of the program and the latter with its application and dynamics.

The basic categories of the formal framework introduced in chapters 2 and 3 aim to represent the grammar in such a way that its application can be effected in an algorithmic manner. The data-structures specified below are developed to render the categories of this formal framework.

5.1 Data-structures

5.1.1 Elements

A collection of elements—components, attributes and meaning-expressions—constitutes the basic database. A typical entry of this database looks as follows:

```
1 paTha_a;
2   ait_9 udAttet_9;
3   p_0;
4   a_0 hrasva_0;
5   Th_1;
6   a_0 hrasva_0 it_0 udAtta_0
```

The above string is separated by semi-colons (;). Here, the first entry (line 1) is the ID of the element. It is the ID of the verbal root paTha_a. After this (line 2) the IDs of its attributes are noted with a blank space in between. There are two IDs mentioned here: ait_9 which stands for the attribute that has the phoneme a_0 as it_0 marker. Secondly, the attribute udAttet_9 says that the marker sound here is high-pitched. This is followed by entries for each phoneme and their attributes. There are four phonemes here (lines 3-6). The final phoneme a_0 carries the attribute it_0 and udAtta_0, that makes the whole component ait_9 and udAttet_9.

The collection of elements is implemented by the `Elements` class.

¹ The Python Programming Language—Official website <https://www.python.org> (accessed on 22.03.2016) supplies free download of the Python interpreter as well as comprehensive documentation.

```

1 class Elements:
2     def __init__(self):
3         self.elements = dictionary_of_elements(ELEMENTS_FILE_NAME)

```

The class variable is a dictionary of elements initialized by the file `ELEMENTS_FILE_NAME`. The class of elements can be instantiated as follows.

```

1 >>> from elements import Elements
2 >>> element = Elements()

```

The first command imports the class of Elements and the second one instantiates it. The variable `element` is now an object variable belonging to the Elements class. The class functions can now be executed. For example, the following function returns the boolean value True or False, depending upon whether an element is present in the database or not.

```

1 >>> element.is_an_element('paTha_a')
2 True

```

Another function of this class returns a component as a list of sets, which can be represented as language-components.

```

1 >>> element.get_component_in_langComp_form('paTha_a')
2 [set(['paTha_a', 'udAttet_9', 'p_0', 'ait_9']),
3  set(['hrasva_0', 'paTha_a', 'udAttet_9', 'a_0', 'ait_9']),
4  set(['paTha_a', 'udAttet_9', 'Th_1', 'ait_9'])]

```

A list consisting of three sets is returned in the above example. These three sets correspond to the three sounds of the said component. It should be noted that certain attributes, like `ait_9` i.e. having a as *it*-marker or `udAttet_9` having an *udatta*-marker are included in the database itself.

The value returned is suitable for representation in the new framework. Thus, the final marker sound is not included as part of the form of the component, but as an attribute. Thus, it has only three sets corresponding to the three sounds /p a ðh/, and not four sets. This is because the final phoneme in the original corpus is only a marker sound.

5.1.2 Sound-sets

Each of the three sets representing the three phonemes of the component `paTha_a` are stored as sound-sets which are implemented by the class `SoundSets`.

```

1 class SoundSets:
2     def __init__(self, a_set_of_item_ids):
3         self.soundSet = a_set_of_item_ids

```

It is instantiated by a set of IDs. For example, the above list of three sets can be represented through three SoundSets.

```

1 >>> from soundSets import SoundSets
2 >>> soundSet_1 = SoundSets(set(['paTha_a', 'udAttet_9',
3   'p_0', 'ait_9']))
4 >>> soundSet_2 = SoundSets(set(['paTha_a', 'udAttet_9',
5   'hrasva_0', 'a_0', 'ait_9']))
6 >>> soundSet_3 = SoundSets(set(['paTha_a', 'udAttet_9',
7   'Th_1', 'ait_9']))

```

A sound-set contains a collection of IDs. There is, however, an important constraint: it must have *exactly one* element from the following set of fundamental sounds.

```

1 FUNDAMENTAL_SOUNDS=['a_0', 'i_0', 'u_0',
2   'R_1', 'lR_0',
3   'e_0', 'o_0',
4   'ai_0', 'au_0',
5   'h_0', 'y_0', 'v_0', 'r_0',
6   'l_0',
7   'J_1', 'm_0', 'G_1', 'N_1', 'n_0',
8   'jh_0', 'bh_0',
9   'gh_0', 'Dh_1', 'dh_0',
10  'j_0', 'b_0', 'g_0', 'D_1', 'd_0',
11  'kh_0', 'ph_0', 'ch_0', 'Th_1', 'th_0', 'c_0', 'T_1', 't_0',
12  'k_0', 'p_0',
13  'z_0', 'S_1', 's_0',
14  'h_0',
15  'H_1', 'M_1',
16  'x_0']

```

In the above list, IDs in the first fourteen lines correspond to the fourteen Śiva-sūtras. In the 15th line, H_1 and M_1 correspond to the aspirated sound *visarjanīya* and the nasal sound *anusvāra* respectively. Finally, x_0 in the 16th line is for a pause or *virāma*.

Because of the constraint that exactly one ID from the above set must be present within a sound-set, the following set of IDs are invalid candidates for a sound-set.

```

1 >>> from soundSets import SoundSets
2 >>> soundSet_4 = SoundSets(set(['paTha_a']))
3 Invalid sound-set!

```

The soundSet_4 does not contain any ID from the above set of FUNDAMENTAL_SOUNDS and hence is an invalid set for a sound-set.

```

1 >>> soundSet_5 = SoundSets(set(['a_0', 'i_0']))
2 Invalid sound-set!

```

Here, soundSet_5 has more than one ID from the set of FUNDAMENTAL_SOUNDS.

The following function of the class returns the phonetic form of a sound-set. The phonetic form may contain additional attributes like the length or intonation of vowels.

```
1 >>> soundSet_1.get_phoneme_as_a_set()
2 set(['p_0'])
3 >>> soundSet_2.get_phoneme_as_a_set()
4 set(['hrasva_0', 'a_0'])
```

A new attribute can be added to a sound-set. The following function implements it.

```
1 >>> soundSet_2.addAttributes(set(['at_1']))
2 >>> soundSet_2.get_soundSet()
3 set(['paTha_a', 'udAttet_9', 'a_0', 'hrasva_0', 'at_1', 'ait_9'])
```

Here, the attribute `at_1` is added to a sound-set. While adding the attributes, the consistency condition of the sound-set is taken care of. Thus, an attribute like `dIrgha_0` can not coexist with `hrasva_0` within the same sound-set.

```
1 >>> soundSet_2.addAttributes(set(['dIrgha_0']))
2 dIrgha_0 cannot coexist with hrasva_0 !!
```

Similarly, existing attributes can be removed from a sound-set.

```
1 >>> soundSet_2.remAttributes(set(['at_1']))
2 >>> soundSet_2.get_soundSet()
3 set(['paTha_a', 'udAttet_9', 'a_0', 'hrasva_0', 'ait_9'])
```

5.1.3 Language-components

Language-components constitute an intermediate unit between the whole sentences and the individual sounds. They may roughly (but not necessarily) correspond to an inflected word within a sentence. From the point of view of data-structures, they are a list of sound-sets. The `LangComps` class implements the language-components.

```
1 class LangComps:
2     def __init__(self, list_of_sets=[]):
3         self.langComp=[SoundSets(ss) for ss in list_of_sets]
```

In order to initialize a language-component, the component should be rendered in a special form, namely as a sequence of sets of IDs. This is achieved by a function from the `Elements` class.

```
1 >>> from elements import Elements
2 >>> element = Elements()
3 >>> paTha=element.get_component_in_langComp_form('paTha_a')
```



```

4 >>> print paTha
5 [set(['paTha_a', 'udAttet_9', 'p_0', 'ait_9']),
6  set(['hrasva_0', 'paTha_a', 'udAttet_9', 'a_0', 'ait_9']),
7  set(['paTha_a', 'udAttet_9', 'Th_1', 'ait_9'])]

```

An object of the class `LangComps` can now be instantiated by using the above rendering of the component `paTha_a`.

```

1 >>> from langComps import LangComps
2 >>> langComp = LangComps(paTha)
3 >>> print langComp
4 p : p_0,ait_9,paTha_a,udAttet_9 *
5 a : hrasva_0,a_0,ait_9,paTha_a,udAttet_9 *
6 Th : Th_1,ait_9,paTha_a,udAttet_9

```

The output shows that the language-component has three sound-sets representing the sounds /p, a, Th/ respectively. The sound-sets contain several other IDs that characterise them as well as the language-component.

A number of functions are required to execute operations on language-components. For example, in order to check the conditions, it is important to identify the range of indices in which some attribute occurs. The question as to which sound-set in a particular language-component has any of the given attributes is implemented by the following function.

```

1 >>> langComp.range_withAny(['hrasva_0'])
2 [1]

```

It says that the attribute `hrasva_0` is in the second sound-set of the current language-component.² In case more than one IDs are searched, then this function returns all indices where *any* of the IDs occur.

```

1 >>> langComp.range_withAny(['a_0', 'p_0'])
2 [0, 1]

```

In the example above, indices of those sound-sets that contain any of the IDs in the list `['a_0', 'p_0']` are returned.

If one wants to attach the attribute `dhAtu_0` to those parts of language-component which contains the ID `paTha_a` then the range of indices with the ID `paTha_a` needs to be identified first, followed by the addition of the attribute `dhAtu_0` to the respective indices.

```

1 >>> langComp.range_withAny(['paTha_a'])
2 [0, 1, 2]
3 >>> langComp.addAttributes(set(['dhAtu_0']), [0,1,2])
4 >>> print langComp
5 p : p_0,ait_9,dhAtu_0,paTha_a,udAttet_9 *
6 a : hrasva_0,a_0,ait_9,dhAtu_0,paTha_a,udAttet_9 *
7 Th : Th_1,ait_9,dhAtu_0,paTha_a,udAttet_9

```

² The lists in Python programming language are indexed beginning with 0. So the first element of a list `list` is `list[0]` and the second one is `list[1]` etc.

The above example shows that the attribute `dhAtu_0` is added to the indices corresponding to the sound-sets that contain `paTha_a`.

5.1.4 Sentences

Sentences consist of one or more language-components. They represent the whole unit of a typical linguistic expression with the possibility of a number of inflected words. This class is necessary since the rules of grammar consider the whole sentence and not just one word to be the unit of derivation.

From the point of view of a formal representation of grammatical processes, sentences can simply be defined as a sequence of language-components. Accordingly, the class of Sentences is implemented as a list of LangComps.

```

1 class Sentences:
2     def __init__(self, list_of_LangComps=[LangComps()]):
3         self.sentence = list_of_LangComps

```

The present formal framework uses three levels to represent any linguistic expression. Sentences correspond to the whole unit of a particular linguistic expression, while sound-sets correspond to the individual sounds. Language-components are an intermediate level between the two and are tentatively related to an inflected word. Depending upon the level from which conditional information can be gathered in a sufficient manner, the grammatical operations can be distinguished as those that apply to a sound-set or to a language-component or at the level of the whole sentence.

5.1.5 Derivational states

The process of derivation is carried out when an operational statement is applied to a sentence or to its constituents, i.e. the language-components or the sound-sets. Together with an operational statement, a sentence forms the next data-structure of the system, namely the class DStates.

```

1 class DStates: # Derivational state
2     def __init__(self, (sentence, statement_string)=(None, None)):
3         self.dState = sentence
4         self.applied_statement_str = statement_string

```

The application of a particular statement brings about some change in the current state of a sentence. This change may be at the level of a sound-set if, for example, it gets a new attribute, or at the level of a language-component,

for example, addition of a new sound-set, or even at the level of a sentence itself, in the case of addition of new language-components. The `dState` variable of the class saves the changed state of the sentence after the application of some statement. The operational statement which is applied is stored in the variable `applied_statement_str` (see section 5.1.8 for the nature of an operational string).

5.1.6 Slices

The current state of a sentence saved in a derivational state is the result of application of an operational statement on the previous state of that sentence. The sequence of such derivational states is stored in a slice and is implemented by the class `Slices`. From the point of view of data-structures, a slice is simply a sequence or list of derivational states or `DStates`.

```
1 class Slices:
2     def __init__(self, list_of_DStates=[]):
3         self.slice = list_of_DStates
```

There are two kinds of changes that the operational rules of grammar bring about: either the derivational state is *saturated* or it progresses towards *completion*. The process of saturation is associated with attachment of attributes, while addition of new components is related with incremental steps of completion of the derivational process. Slices contain only those changes where the derivational process is saturated, i.e. only when attributes are added to the components. The other case, when a new component is introduced, leads to the formation of a new slice.

5.1.7 Process-strips

The incremental steps of completion of the process of derivation results in a sequence of slices. Process-strips record this sequences of slices. This is implemented through the class `PStrips`.

```
1 class PStrips: # process-strips
2     def __init__(self, list_of_Slices=[]):
3         self.pStrip=list_of_Slices
```

The data-structures introduced thus far adequately represent the constituents and processes of the grammar in an integral manner. The processes of grammar are enacted through the operational rules. The next data-structure comprehends them.

5.1.8 Statements

In the previous chapter, the concept of statements in comparison with the *sūtras* was introduced and specified. Statements are operational rules that are formulated in a formal framework and can be implemented through algorithmic functions.

The information related to a statement is stored in a particular format within the database. In order to point out the structure of the database of statements, consider the entry corresponding to the attachment of the attribute `vRddhi_0`. In the original corpus it is the very first *sūtra*.

```

1 Xm_ATT_a *
2 ATT_a & vRddhi_0 *
3 Xm:[At_2,aic_0] AND Xm_NOT:[vRddhi_0] *
4 ST_TYPE:[STABILIZING] *
5 A_RULES:[a_11001]
```

The above entry consists of five parts that are separated by a * sign. Each part is listed here in a separate line.

1. The first part denotes the type of the statement. In the present case, it is given by `Xm_ATT_a`. It implies that this statement is about ATTachment of an attribute `a` to some sound-set `Xm`.
2. The second part specifies the operation. In the present case it is given by `ATT_a & vRddhi_0`. It implies that `vRddhi_0` is the ATTachment here.
3. The third part notes the conditions that should be fulfilled in order to execute the operation. There are two parts and both of them need to be fulfilled. Hence they are conjoined by the logical AND.
 - a. The first part of the condition is given by `Xm:[At_2,aic_0]` that can be interpreted as the presence of either the attribute `At_2` or the attribute `aic_0` in the sound-set `Xm`.
 - b. The second part `Xm_NOT:[vRddhi_0]` ensures that the attribute is attached only if it is not already included in the said sound-set. This is important to avoid recursive attachment of an attribute.
4. The inter-relations between other statements within the database are noted in the fourth part. It also records the nature of the statements. Here, for example, it says that this statement is a STABILIZING statement which contributes to the *saturation* of the slice.
5. Finally, the fifth part records the links to the external associations, especially the correspondence with the original corpus of the Aṣṭādhyāyī. Here, the *sūtra* number `a_11001` is noted.

An individual operational statement is implemented by the `Statements` class and is instantiated by the corresponding entry string within the database.

```

1 class Statements:
2     def __init__(self, st_str=''):
3         self.statement_string=st_str
4         self.statement = parse_a_statement_str(st_str)

```

The function `parse_a_statement_str(st_str)` parses and returns the output as a five-tuple for the instantiation variable.

```

1 >>> statement_string = 'Xm_ATT_a *
2     ATT_a & vRddhi_0 *
3     Xm:[At_2,aic_0] AND Xm_NOT:[vRddhi_0] *
4     ST_TYPE:[STABILIZING] *
5     A_RULES:[a_11001]'
6 >>> from statements import Statements
7 >>> statement = Statements(statement_string)

```

This class implements several functions that are important for the application of the statements. One such function is to get the *signature* of a given statement.

```

1 >>> statement.get_signature()
2     'Xm_ATT_a__ATT_a__Xm_Xm_NOT'

```

The signature of a statement specifies its *structure*. Associated with a statement, there is a function which executes its application. The nature of this applicational function is defined by the signature of the statement. All statements with the same signature can be applied by using the same function.

Another function supplies the information about the conditions that need to be fulfilled in order to apply that statement.

```

1 >>> statement.get_condition_type_vals_dict()
2     {'Xm': [['At_2', 'aic_0']],
3     'Xm_NOT': [['vRddhi_0']]}

```

Here, the return value is a dictionary. Its keys are the types or nature of the conditions together with the corresponding values for the particular case. Thus, the first condition type is `Xm` implying that the sound-set must contain any of the IDs `At_2` or `aic_0`. The type of the second condition is `Xm_NOT` and it says that the ID `vRddhi_0` should not be in that sound-set.

5.1.9 Statement groups

The collection of `Statements` is implemented through the `StatementGroups` class.

```

1 class StatementGroups:
2     def __init__(self):
3         self.statementGroup = list_of_Statements

```

The functions of this class are primarily meant for organizing and referencing the statements.

```

1 >>> from statements import StatementGroups
2 >>> statementGroup = StatementGroups()
3 >>> statementGroup.get_statements_for_some_operation(
4     'ATT_a & guru_0')
5     [<statements.Statements instance at 0x10a24c3f8>,
6      <statements.Statements instance at 0x10a24c488>]

```

The list returned by the above function consists of two Statements instances corresponding to the two statements that provide for application of the attribute guru_0.

```

1 >>> st1,st2=statementGroup.get_statements_for_some_operation(
2     'ATT_a & guru_0')
3 >>> st1.statement_string
4     'Xm_ATT_a * ATT_a & guru_0 * Xm:[hrasva_0] AND Xn:[saMyoga_0]*
5     ST_TYPE:[STABILIZING] * A_RULES:[a_14010][a_14011]'
6 >>> st2.statement_string
7     'Xm_ATT_a * ATT_a & guru_0 * Xm:[dIrga_0] *
8     ST_TYPE:[STABILIZING] * A_RULES:[a_14012]'

```

Similarly, the following function returns a dictionary with signature of statements as keys and the corresponding statements as their values.

```

1 >>> d=statementGroup.get_signature_statements_dict()
2 >>> d
3 {'Xm_ATT_a__ATT_a__Xm_Xm_NOT': [
4     <statements.Statements instance at 0x10a382638>,
5     <statements.Statements instance at 0x10a382950>,
6     <statements.Statements instance at 0x10a382998>,
7     ...], ... }

```

5.2 Processes of grammar

During the process of their derivation, linguistic expressions are represented through a sentence which consists of one or more language-components corresponding to the individual inflected words. Each language-component contains a sequence of sound-sets. Each sound-set corresponds to a single phoneme.

The derivational process is effected through a number of operational statements which are applied to a sentence. A sentence, together with an opera-

tional statement, constitutes a derivational state. There are two fundamental types of operations: (i) to saturate a sentence, in that all attributes that can be attached are added to it, and (ii) to add a new component and graduate towards completion of the derivational process. Accordingly, a slice contains a sequence of derivational states that arise during the process of saturation. A new slice is added, once a new component is introduced. A process-strip records a sequence of slices and thus registers the process of completion.

Given the above framework and corresponding data-structures, the general algorithm of the derivational process can be specified as follows.

```

1 initialize a process-strip
2 repeat the following steps:
3     saturate the process-strip
4     look for completing statements
5     if there is no statement to be applied
6         return the process-strip
7     else:
8         select a completing statement
9         apply it to the process-strip

```

After initialisation, the process-strip is populated with new components and saturated repeatedly, till there is no admissible component available. This brings the process of derivation to an end.

5.2.1 Initialisation

The process of initialisation is implemented by the function `initialize()`. The initial process-strip contains an empty slice.

```

1 >>> from pStrips import PStrips
2 >>> pStrip = PStrips()
3 >>> pStrip.get_list_of_Slices()
4 []

```

The empty `pStrip` needs to be populated with some components. At this moment the meaning-expressions and a user become relevant. The user must be able to express her / his intention (*vivakṣā*) by interacting with the system. Suppose, for example, the user wants to express the sentence *bālakaḥ paṭhati* (a boy recites). Then the choice of the components `bAlaka_k` and `paTha_a` become imminent.³ The following statement is chosen for application.

```

1 >>> st01 = Statements(
2     'E_ADD_y *

```

³ For the sake of simplicity and space, I will continue with the derivation of the verbal conjugation only.

```

3   ADD_y & paTha_a *
4   yM:[vyaktAyAM_x vAci_x] *
5   ST_TYPE:[COMPLETING] * ')
6 >>> st01.get_signature()
7   'E_ADD_y__ADD_y__yM'

```

The signature of the above statement is significant for choosing the appropriate signature-functions. These are required to execute some statement. Consider the following function for the application of statements with signature: E_ADD_y__ADD_y__yM.

```

1 def E_ADD_y__ADD_y__yM(pStrip, statement):
2     (op_type, op_val) = statement.get_operation_part()
3     list_of_sets=Elements().get_component_in_langComp_form(op_val)
4     langComp = LangComps(list_of_sets)
5     sentence = Sentences(
6         ([langComp], statement.get_vals_for_condition_type('yM')))
7     dState = DStates((sentence, statement.get_statement_string()))
8     slice = Slices([dState])
9     pStrip = PStrips([slice])
10    return pStrip

```

[1] The function takes pStrip and statement objects and returns the updated pStrip after application of the statement. [2] The values of operation-type and operation-value are stored in the variables op_type and op_val respectively. These are 'ADD_y' and 'paTha_a'. [3] The variable list_of_sets contains the list of sets of IDs corresponding to the op_val which in this case is 'paTha_a'. [4] The langComp is instantiated, followed by [5] sentence, [7] dState, [8] slice and [9] pStrip. [10] The updated pStrip is returned.

The application of the above signature function results in the introduction of a new slice within the process-strip. Within this slice a new derivational state is added which records the changes in the sentence and saves the statement that is applied as well. The results are as follows.

```

1 >>> from signatureFunctions import E_ADD_y__ADD_y__yM
2 >>> pStrip = E_ADD_y__ADD_y__yM(pStrip, st01)
3 >>> print pStrip
4 p : p_0,ait_9,paTha_a,udAttet_9 *
5 a : hrasva_0,a_0,ait_9,paTha_a,udAttet_9 *
6 Th : Th_1,ait_9,paTha_a,udAttet_9
7 :-: E_ADD_y * ADD_y & paTha_a *
8     yM:[vyaktAyAM_x vAci_x] *
9     ST_TYPE:[COMPLETING] *

```


5.2.2 Saturation

At this stage the process of saturation is carried out. Consider the following statement.

```

1 >>> st02 = Statements(
2     'Xm_ATT_a *
3     ATT_a & at_1 *
4     Xm:[a_0][hrasva_0] AND Xm_NOT:[at_1] *
5     ST_TYPE:[STABILIZING] *
6     A_RULES:[a_11070]')
```

This statement attaches the attribute `at_1` to a sound-set `Xm` that fulfils the following two conditions:

1. `Xm:[a_0][hrasva_0]` i.e. the sound-set `Xm` must contain the phoneme `a_0` and the attribute `hrasva_0`.
2. `Xm_NOT:[at_1]` implies that the attribute should not already be present in the said sound-set. This is necessary to avoid recursive attachment of an attribute.

The signature of this statement is as follows.

```

1 >>> st02.get_signature()
2     'Xm_ATT_a__ATT_a__Xm_Xm_NOT'
```

The implementation of this statement is effected through the following signature-function.

```

1 def Xm_ATT_a__ATT_a__Xm_Xm_NOT(pStrip, statement):
2     sentence = _get_a_deepcopy_of_sentence(pStrip)
3     for langComp in sentence.sentence:
4         for soundSet in langComp.langComp:
5             chk_results_dict = _Xm_ATT_a__ATT_a__Xm_Xm_NOT__CHECK(
6                 soundSet, statement)
7             if chk_results_dict.get('APPLICABILITY'):
8                 soundSet.addAttributes(
9                     set([chk_results_dict.get('ATT_a')]))
10                pStrip.get_last_Slice().extend_Slice(
11                    DStates((sentence, statement.get_statement_string())))
12     return pStrip
```

[1] Again the function takes up the `pStrip` and a statement and [12] returns the updated strip after the application of the statement. [2] A deep copy of the sentence is needed to avoid over writing. [3-4] Since the operation is executed at the level of sound-sets, the two `for` loops are carried out. [5-6] The function `..._CHECK` checks the conditions whether the statement is applicable to the `soundSet` or not. The results of this check function are stored in the dictionary `chk_results_dict`. [7] If applicable, then [8-9] the appropriate attribute is added to that `soundSet`, and [10-11] `pStrip` gets updated.

The above function uses another function to check the conditions.

```

1 def _Xm_ATT_a__ATT_a__Xm_Xm_NOT__CHECK(soundSet, statement):
2     chk_results_dict = {}
3     oper_part_key, oper_part_val = statement.get_operation_part()
4     if CHK_Xm_Xm_NOT(soundSet, statement):
5         chk_results_dict['APPLICABILITY'] = True
6         chk_results_dict[oper_part_key] = oper_part_val
7     return chk_results_dict

```

[1] The `..._CHECK` function takes a `soundSet` and a `statement` and returns `chk_results_dict` a dictionary of results. [4] It uses another function `CHK_Xm_Xm_NOT` that checks whether the `Xm` and `Xm_NOT` conditions are fulfilled.

The statement `st02` can now be applied.

```

1 >>> from signatureFunctions import Xm_ATT_a__ATT_a__Xm_Xm_NOT
2 >>> pStrip = Xm_ATT_a__ATT_a__Xm_Xm_NOT(pStrip, st02)
3 >>> print pStrip
4 p : p_0,ait_9,paTha_a,udAttet_9 *
5 a : hrasva_0,a_0,ait_9,paTha_a,udAttet_9 *
6 Th : Th_1,ait_9,paTha_a,udAttet_9
7 :-: E_ADD_y * ADD_y & paTha_a *
8     yM:[vyaktAyAM_x vAci_x] *
9     ST_TYPE:[COMPLETING] *
10 :::
11 p : p_0,ait_9,paTha_a,udAttet_9 *
12 a : hrasva_0,a_0,ait_9,at_1,paTha_a,udAttet_9 *
13 Th : Th_1,ait_9,paTha_a,udAttet_9
14 :-: Xm_ATT_a * ATT_a & at_1 *
15     Xm:[a_0][hrasva_0] AND Xm_NOT:[at_1] *
16     ST_TYPE:[STABILIZING] * A_RULES:[a_11070]

```

The application of the above statement has resulted in an extension of the slice. [10] A new derivational state is added. [12] The effect of this function is visible in this line, where the sound-set gets the attribute `at_1` added to it. This is the only sound-set where the conditions of the statement are fulfilled.

It should be noted that multiple application of this statement to the same `pStrip` does not result in any further changes.

```

1 >>> pStrip = Xm_ATT_a__ATT_a__Xm_Xm_NOT(pStrip, st02)
2 >>> pStrip = Xm_ATT_a__ATT_a__Xm_Xm_NOT(pStrip, st02)
3 >>> print pStrip
4 p : p_0,ait_9,paTha_a,udAttet_9 *
5 a : hrasva_0,a_0,ait_9,paTha_a,udAttet_9 *
6 Th : Th_1,ait_9,paTha_a,udAttet_9
7 :-: E_ADD_y * ADD_y & paTha_a *
8     yM:[vyaktAyAM_x vAci_x] *
9     ST_TYPE:[COMPLETING] *
10 :::
11 p : p_0,ait_9,paTha_a,udAttet_9 *

```

```

12 a : hrasva_0,a_0,ait_9,at_1,paTha_a,udAttet_9 *
13 Th : Th_1,ait_9,paTha_a,udAttet_9
14 :-: Xm_ATT_a * ATT_a & at_1 *
15     Xm:[a_0][hrasva_0] AND Xm_NOT:[at_1] *
16     ST_TYPE:[STABILIZING] * A_RULES:[a_11070]

```

Consider now the following two statements.

```

1 >>> st03 = Statements(
2     'Xm_ATT_a *
3     ATT_a & vRddhi_0 *
4     Xm:[At_2,aic_0] AND Xm_NOT:[vRddhi_0] *
5     ST_TYPE:[STABILIZING] * A_RULES:[a_11001]')
6 >>> st04 = Statements(
7     'Xm_ATT_a *
8     ATT_a & guNa_0 *
9     Xm:[at_1,eG_0] AND Xm_NOT:[guNa_0] *
10    ST_TYPE:[STABILIZING] * A_RULES:[a_11002]')

```

Both of them have the same signature as the previous statement.

```

1 >>> st03.get_signature()
2     'Xm_ATT_a__ATT_a__Xm_Xm_NOT'
3 >>> st04.get_signature()
4     'Xm_ATT_a__ATT_a__Xm_Xm_NOT'

```

This implies that both of them can be applied by using the same signature function. The execution of st03 shows that it is not applied at all. This is obvious as none of the sound-sets fulfill the required conditions. The pStrip remains as it is. This is evident from the output below.

```

1 >>> pStrip = Xm_ATT_a__ATT_a__Xm_Xm_NOT(pStrip,st03)
2 >>> print pStrip
3 p : p_0,ait_9,paTha_a,udAttet_9 *
4 a : hrasva_0,a_0,ait_9,paTha_a,udAttet_9 *
5 Th : Th_1,ait_9,paTha_a,udAttet_9
6 :-: E_ADD_y * ADD_y & paTha_a *
7     yM:[vyaktAyAM_x vAci_x] *
8     ST_TYPE:[COMPLETING] *
9     :::
10 p : p_0,ait_9,paTha_a,udAttet_9 *
11 a : hrasva_0,a_0,ait_9,at_1,paTha_a,udAttet_9 *
12 Th : Th_1,ait_9,paTha_a,udAttet_9
13 :-: Xm_ATT_a * ATT_a & at_1 *
14     Xm:[a_0][hrasva_0] AND Xm_NOT:[at_1] *
15     ST_TYPE:[STABILIZING] * A_RULES:[a_11070]

```

The execution of st04 however brings out some changes.

```

1 >>> pStrip = Xm_ATT_a__ATT_a__Xm_Xm_NOT(pStrip,st04)
2 >>> print pStrip
3 p : p_0,ait_9,paTha_a,udAttet_9 *
4 a : hrasva_0,a_0,ait_9,paTha_a,udAttet_9 *

```

```

5 Th : Th_1,ait_9,paTha_a,udAttet_9
6 :-: E_ADD_y * ADD_y & paTha_a *
7     yM:[vyaktAyAM_x vAci_x] *
8     ST_TYPE:[COMPLETING] *
9     :::
10 p : p_0,ait_9,paTha_a,udAttet_9 *
11 a : hrasva_0,a_0,ait_9,at_1,paTha_a,udAttet_9 *
12 Th : Th_1,ait_9,paTha_a,udAttet_9
13 :-: Xm_ATT_a * ATT_a & at_1 *
14     Xm:[a_0][hrasva_0] AND Xm_NOT:[at_1] *
15     ST_TYPE:[STABILIZING] * A_RULES:[a_11070]
16     :::
17 p : p_0,ait_9,paTha_a,udAttet_9 *
18 a : hrasva_0,a_0,ait_9,at_1,guNa_0,paTha_a,udAttet_9 *
19 Th : Th_1,ait_9,paTha_a,udAttet_9
20 :-: Xm_ATT_a * ATT_a & guNa_0 *
21     Xm:[at_1,eG_0] AND Xm_NOT:[guNa_0] *
22     ST_TYPE:[STABILIZING] * A_RULES:[a_11002]

```

[18] The attribute `guNa_0` is now attached to the sound-set having the attribute `at_1`. There is a hierarchy among the saturating statements that is decided by the dependency of their conditions together with the operational attachment. Thus, while `st03` and `st04` can be applied in any order, `st04` can only be applied after the application of `st02`.

The next statement attaches an attribute not only to just one sound-set, but to several of them.

```

1 >>> st05 = Statements(
2     'X_ATT_a *
3     ATT_a & bhvAdi_0 *
4     X:[bhU_a,paTha_a,ji_a,pUG_a,dheT_a] AND X_NOT:[bhvAdi_0] *
5     ST_TYPE:[STABILIZING] * ')
6 >>> st05.get_signature()
7     'X_ATT_a__ATT_a__X_X_NOT'

```

This statement attaches the group name `bhvAdi_0` for the *bhvādi-gaṇa* of the verbal roots. Here, for the sake of readability, I have not listed all the roots that are mentioned in this group.

```

1 >>> from signatureFunctions import X_ATT_a__ATT_a__X_X_NOT
2 >>> pStrip = X_ATT_a__ATT_a__X_X_NOT(pStrip,st05)
3 >>> print pStrip
4     ...
5     :::
6 p : p_0,ait_9,bhvAdi_0,paTha_a,udAttet_9 *
7 a : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,guNa_0,paTha_a,
8     udAttet_9 *
9 Th : Th_1,ait_9,bhvAdi_0,paTha_a,udAttet_9
10 :-: X_ATT_a * ATT_a & bhvAdi_0 *
11     X:[bhU_a,paTha_a,ji_a,pUG_a,dheT_a] AND X_NOT:[bhvAdi_0] *
12     ST_TYPE:[STABILIZING] *

```

The attribute `bhvAdi_0` is added to all three of the sound-sets that constitute the component `paTha_a`. In the above output, I have deleted the earlier derivational states to save space and to aid readability.

The next statement that attaches the attribute `dhAtu_0` is similar.

```

1 >>> st06 = Statements(
2   'X_ATT_a *
3   ATT_a & dhAtu_0 *
4   X:[bhvAdi_0,adAdi_0,juhotyAdi_0,divAdi_0,svAdi_0,
5     tudAdi_0,rudhAdi_0,tanAdi_0,kryAdi_0,curAdi_0] AND
6   X_NOT:[dhAtu_0] *
7   ST_TYPE:[STABILIZING] * A_RULES:[a_13001]')
8 >>> st06.get_signature()
9 'X_ATT_a__ATT_a__X_X_NOT'
10 >>> from signatureFunctions import X_ATT_a__ATT_a__X_X_NOT
11 >>> pStrip = X_ATT_a__ATT_a__X_X_NOT(pStrip,st06)
12 >>> print pStrip
13 ...
14 :::
15 p : p_0,ait_9,bhvAdi_0,dhAtu_0,paTha_a,udAttet_9 *
16 a : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,dhAtu_0,guNa_0,
17   paTha_a,udAttet_9 *
18 Th : Th_1,ait_9,bhvAdi_0,dhAtu_0,paTha_a,udAttet_9
19 :-: X_ATT_a * ATT_a & dhAtu_0 *
20   X:[bhvAdi_0,adAdi_0,juhotyAdi_0,divAdi_0,svAdi_0,
21     tudAdi_0,rudhAdi_0,tanAdi_0,kryAdi_0,curAdi_0] AND
22   X_NOT:[dhAtu_0] *
23   ST_TYPE:[STABILIZING] * A_RULES:[a_13001]

```

5.2.3 Completion

The next statement introduces the component `laT_0`. This component is added *after* the language-component `Xi`. The semantic condition for its addition is stated in the condition `yM`: `[vartamAna_x]`. This condition is satisfied once the user confirms her or his intention to express `vartamAna_x` or present tense.

```

1 >>> st07 = Statements(
2   'Xi_ADD_y *
3   ADD_y & laT_0 *
4   Xi:[dhAtu_0] AND Xj_NOT:[lakAra_9] AND yM:[vartamAna_x] *
5   ST_TYPE:[COMPLETING] * A_RULES:[a_32123]')
6 >>> st07.get_signature()
7 'Xi_ADD_y__ADD_y__Xi_Xj_NOT_yM'

```

The above statement is applied through the corresponding signature function. It is a completing statement and therefore a new slice is added. Line [16] notes the slice boundary.

```

1 >>> from signatureFunctions import Xi_ADD_y__ADD_y__Xi_Xj_NOT_yM
2 >>> pStrip = Xi_ADD_y__ADD_y__Xi_Xj_NOT_yM(pStrip,st07)
3 >>> print pStrip
4 ...
5 :::
6 p : p_0,ait_9,bhvAdi_0,dhAtu_0,paTha_a,udAttet_9 *
7 a : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,dhAtu_0,guNa_0,
8     paTha_a,udAttet_9 *
9 Th : Th_1,ait_9,bhvAdi_0,dhAtu_0,paTha_a,udAttet_9
10 :-: X_ATT_a *
11     ATT_a & dhAtu_0 *
12     X:[bhvAdi_0,adAdi_0,juhotyAdi_0,divAdi_0,svAdi_0,
13     tudAdi_0,rudhAdi_0,tanAdi_0,kryAdi_0,curAdi_0] AND
14     X_NOT:[dhAtu_0] *
15     ST_TYPE:[STABILIZING] * A_RULES:[a_13001]
16 -:-::-
17 p : p_0,ait_9,bhvAdi_0,dhAtu_0,paTha_a,udAttet_9 *
18 a : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,dhAtu_0,guNa_0,
19     paTha_a,udAttet_9 *
20 Th : Th_1,ait_9,bhvAdi_0,dhAtu_0,paTha_a,udAttet_9 *
21 l : l_0,Tit_9,ait_9,laT_0,lakAra_9
22 :-: Xi_ADD_y *
23     ADD_y & laT_0 *
24     Xi:[dhAtu_0] AND Xj_NOT:[lakAra_9] AND yM:[vartamAna_x] *
25     MEXGRP_0001 * A_RULES:[a_32123]

```

[21] The above statement also extends the language-component and a new sound-set for laT_0 is added. laT_0 is a Tit_9 and ait_9 lakAra_9 and these attributes are already specified in the database.

The next statement specifies whether the components within a particular language-component form a sentence in active, passive or middle voice. Again, the decision to employ active voice is reached on the basis of the semantic condition, which the user must address directly.

```

1 >>> st08 = Statements(
2     'X_ATT_a *
3     ATT_a & xkartR_9 *
4     X:[dhAtu_0] AND X_NOT:[xkarman_9][xbhAva_9][xkartR_9] AND
5     X_M:[kartRprayoga_x] *
6     ST_TYPE:[STABILIZING] *')
7 >>> st08.get_signature()
8 'X_ATT_a__ATT_a__X_X_M_X_NOT'
9 >>> from signatureFunctions import X_ATT_a__ATT_a__X_X_M_X_NOT
10 >>> pStrip = X_ATT_a__ATT_a__X_X_M_X_NOT(pStrip,st08)
11 >>> print pStrip
12 :::
13 p : p_0,ait_9,bhvAdi_0,dhAtu_0,paTha_a,udAttet_9,xkartR_9 *
14 a : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,dhAtu_0,guNa_0,
15     paTha_a,udAttet_9,xkartR_9 *
16 Th : Th_1,ait_9,bhvAdi_0,dhAtu_0,paTha_a,udAttet_9,
17     xkartR_9 *
18 l : l_0,Tit_9,ait_9,laT_0,lakAra_9

```

```

19  :-: X_ATT_a *
20      ATT_a & xkartR_9 *
21      X:[dhAtu_0] AND X_NOT:[xkarman_9][xbhAva_9][xkartR_9] AND
22      X_M:[kartRprayoga_x] *
23      ST_TYPE:[STABILIZING] *

```

The following statement attaches the attribute `parasmaipada_0` to the language-component.

```

1  >>> st09 = Statements(
2      'X_ATT_a *
3      ATT_a & parasmaipada_0 *
4      X:[dhAtu_0][xkartR_9] AND X_NOT:[Atmanepada_0] *
5      ST_TYPE:[STABILIZING] *')
6  >>> st09.get_signature()
7      'X_ATT_a__ATT_a__X_X_NOT'
8  >>> from signatureFunctions import X_ATT_a__ATT_a__X_X_NOT
9  >>> pStrip = X_ATT_a__ATT_a__X_X_NOT(pStrip,st09)
10 >>> print pStrip
11 ...
12 :::
13 p : p_0,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
14     udAttet_9,xkartR_9 *
15 a : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,dhAtu_0,guNa_0,
16     paTha_a,parasmaipada_0,udAttet_9,xkartR_9 *
17 Th : Th_1,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
18     udAttet_9,xkartR_9 *
19 l : l_0,Tit_9,ait_9,laT_0,lakAra_9
20 :-: X_ATT_a *
21     ATT_a & parasmaipada_0 *
22     X:[dhAtu_0][xkartR_9] AND X_NOT:[Atmanepada_0] *
23     ST_TYPE:[STABILIZING] *

```

The following statement provides for the substitution of `laT_0` by the third person singular suffix `tip_0`.

```

1  >>> st10 = Statements(
2      'Xi_REP_y *
3      REP_y & tip_0 *
4      Xh:[dhAtu_0][parasmaipada_0] AND Xi:[lakAra_9] AND
5      yM:[prathama_0][ekavacana_0] *
6      ST_TYPE:[COMPLETING] * A_RULES:[a_34077][a_34078]')
7  >>> ST10_obj.get_signature()
8      'Xi_REP_y__REP_y__Xh_Xi_yM'

```

Again, the decision to opt for the third person and singular is taken by the user on the basis of the semantic conditions `prathama_0` and `ekavacana_0` respectively.

```

1  >>> from signatureFunctions import Xi_REP_y__REP_y__Xh_Xi_yM
2  >>> pStrip = Xi_REP_y__REP_y__Xh_Xi_yM(pStrip,st10)
3  >>> print pStrip
4  -::-:-

```

```

5 p : p_0,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
6   udAttet_9,xkartR_9 *
7 a : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,dhAtu_0,guNa_0,
8   paTha_a,parasmaipada_0,udAttet_9,xkartR_9 *
9 Th : Th_1,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
10  udAttet_9,xkartR_9 *
11 l : l_0,REPLACED_9,Tit_9,ait_9,laT_0,lakAra_9 *
12 t : t_0,pit_9,tip_0 *
13 i : hrasva_0,i_0,pit_9,tip_0
14 :-: Xi_REP_y *
15     REP_y & tip_0 *
16     Xh:[dhAtu_0][parasmaipada_0] AND Xi:[lakAra_9] AND
17     yM:[prathama_0][ekavacana_0] *
18     ST_TYPE:[COMPLETING] * A_RULES:[a_34077][a_34078]

```

Replacement is implemented by attaching the attribute REPLACED_9 to those parts that are replaced (line [11]) and adding the replacement components at the appropriate index (line [12-13]).

The following statement attaches the attribute tiG_0 to the third person singular suffix tip_0.

```

1 >>> st11 = Statements(
2   'X_ATT_a *
3   ATT_a & tiG_0 *
4   X:[tip_0,tas_0,jhi_0,sip_0,thas_0,tha_0,mip_0,vas_0,mas_0,
5   ta_0,AtAm_0,jha_0,thAs_1,AthAm_0,dhvam_0,iT_1,vahi_0,mahiG_0]
6   AND
7   X_NOT:[tiG_0] *
8   ST_TYPE:[STABILIZING] * A_RULES:[a_34078][a_11071]')
9 >>> st11.get_signature()
10 >>> from signatureFunctions import X_ATT_a__ATT_a__X_X_NOT
11 >>> pStrip = X_ATT_a__ATT_a__X_X_NOT(pStrip,st11)
12 >>> print pStrip
13 :::
14 p : p_0,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
15   udAttet_9,xkartR_9 *
16 a : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,dhAtu_0,guNa_0,
17   paTha_a,parasmaipada_0,udAttet_9,xkartR_9 *
18 Th : Th_1,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
19   udAttet_9,xkartR_9 *
20 l : l_0,REPLACED_9,Tit_9,ait_9,laT_0,lakAra_9 *
21 t : t_0,pit_9,tiG_0,tip_0 *
22 i : hrasva_0,i_0,pit_9,tiG_0,tip_0
23 :-: X_ATT_a *
24     ATT_a & tiG_0 *
25     X:[tip_0,tas_0,jhi_0,sip_0,thas_0,tha_0,mip_0,vas_0,mas_0,
26     ta_0,AtAm_0,jha_0,thAs_1,AthAm_0,dhvam_0,iT_1,vahi_0,
27     mahiG_0] AND
28     X_NOT:[tiG_0] *
29     ST_TYPE:[STABILIZING] * A_RULES:[a_34078][a_11071]

```


To a `tiG_0` or a `zit_9` component, the attribute `sArvadhAtuka_0` is attached by the following statement.

```

1 >>> st12 = Statements(
2     'X_ATT_a *
3     ATT_a & sArvadhAtuka_0 *
4     X:[tiG_0,zit_9] AND X_NOT:[sArvadhAtuka_0] *
5     ST_TYPE:[STABILIZING] * A_RULES:[a_34113]')
6 >>> st12.get_signature()
7     'X_ATT_a__ATT_a__X_X_NOT'
```

The application of this statement is similar to the other attachments of the attributes.

```

1 >>> from signatureFunctions import X_ATT_a__ATT_a__X_X_NOT
2 >>> pStrip = X_ATT_a__ATT_a__X_X_NOT(pStrip,st12)
3 >>> print pStrip
4     ...
5     :::
6 p   : p_0,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
7       udAttet_9,xkartR_9 *
8 a   : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,dhAtu_0,guNa_0,
9       paTha_a,parasmaipada_0,udAttet_9,xkartR_9 *
10 Th : Th_1,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
11       udAttet_9,xkartR_9 *
12 l   : l_0,REPLACED_9,Tit_9,ait_9,laT_0,lakAra_9 *
13 t   : t_0,pit_9,sArvadhAtuka_0,tiG_0,tip_0 *
14 i   : hrasva_0,i_0,pit_9,sArvadhAtuka_0,tiG_0,tip_0
15 :-: X_ATT_a *
16     ATT_a & sArvadhAtuka_0 *
17     X:[tiG_0,zit_9] AND X_NOT:[sArvadhAtuka_0] *
18     ST_TYPE:[STABILIZING] * A_RULES:[a_34113]
```

Finally, the infix `zap_0` is introduced by the following statement.

```

1 >>> st13 = Statements(
2     'Xi_ADD_y *
3     ADD_y & zap_0 *
4     Xi:[dhAtu_0] AND Xj:[sArvadhAtuka_0] AND Xj_M:[kartR_0] * *
5     A_RULES:[a_31068]')
6 >>> st13.get_signature()
7     'Xi_ADD_y__ADD_y__Xi_Xj_Xj_M'
```

The conditions also provide the index where the new components should properly be added.

```

1 >>> from signatureFunctions import Xi_ADD_y__ADD_y__Xi_Xj_Xj_M
2 >>> pStrip = Xi_ADD_y__ADD_y__Xi_Xj_Xj_M(pStrip,st13)
3 >>> print pStrip
4     ...
5     -:-:-:-
6 p   : p_0,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
7       udAttet_9,xkartR_9 *
8 a   : hrasva_0,a_0,ait_9,at_1,bhvAdi_0,dhAtu_0,guNa_0,
```

```

9      paTha_a,parasmaipada_0,udAttet_9,xkartR_9 *
10 Th : Th_1,ait_9,bhvAdi_0,dhAtu_0,paTha_a,parasmaipada_0,
11      udAttet_9,xkartR_9 *
12 a  : hrasva_0,a_0,pit_9,zap_0,zit_9 *
13 l  : l_0,REPLACED_9,Tit_9,ait_9,laT_0,lakAra_9 *
14 t  : t_0,pit_9,sArvadhAtuka_0,tiG_0,tip_0 *
15 i  : hrasva_0,i_0,pit_9,sArvadhAtuka_0,tiG_0,tip_0
16 :-: Xi_ADD_y *
17      ADD_y & zap_0 *
18      Xi:[dhAtu_0] AND Xj:[sArvadhAtuka_0] AND Xj_M:[kartR_0] * *
19      A_RULES:[a_31068]

```

The above steps demonstrate the dynamics of the process of derivation. At each step, a larger number of characterising statements are employed. The main algorithm of the derivational process is summarized in the following main function.

```

1  def execute():
2      pStrip = initialize()
3      while 1:
4          pStrip = saturate(pStrip)
5          possible_statements = interpret(pStrip)
6          if len(possible_statements) == 0:
7              return pStrip
8          else:
9              statement = select(pStrip,possible_statements)
10             pStrip = apply(pStrip,statement)

```

[1] The main function which returns a process-strip. [2] The function `initialize()` initializes the process-strip. [3] The third line specifies a loop. [4] At this stage, a given process-strip is saturated, i.e. attributes are attached to the language-components or sound-sets. [5] Once the process-strip is saturated, it is tested for any possible completing statement that could be applied. The list of all such candidates is stored in `possible_statements`. [6] In case the list is empty, i.e. there is no statement that may be applied, [7] then the process-strip is returned. Otherwise, [9] one statement is selected, depending upon the semantic considerations and the intention (*vivakṣā*) of the user. [10] Finally, that statement is applied and the process-strip is updated.

Appendix A

The Systematic Approach

In this appendix I intend to work out the systematic approach followed by the ancillary disciplines associated with the Vedas.¹ In particular, I examine the Ṛgvedaprātiśākhya,² the Baudhāyanaśulbasūtra,³ and the Aṣṭādhyāyī.⁴ My emphasis is to look into the formal aspects of their basic techniques. The chief outcome is the certainty that they follow a common methodology. Further, it leads to the determination and specification of the details of this systematic approach. The results are of significance as they provide the basis for a formal perspective on the Aṣṭādhyāyī of Pāṇini.

In the following, I briefly mention a conceptual plan for the subsequent investigations, including the new terms introduced by me. I shall use typewriter font whenever I want to emphasize their terminological character. These terms are needed to formulate the details of the systematic approach. They reflect the common character of the content and processes of these disciplines, and are needed to avoid the imposition of specific terms from any particular tradition with their established denotations for the general concepts. Their introduction is all the more necessary as the present study aims to point out features spanning across several disciplines.

The starting point is the observation that the above mentioned texts aim towards retention of a given phenomenon. By a given phenomenon I

¹ 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). For a summary of the literature on phonetics and grammar, see (Scharfe 1977), on ritual practices (Gonda 1977) and for astronomy, see (Pingree 1981).

² For our study, I use the Ṛgvedaprātiśākhya with the commentary of Uvaṭa edited by Virendrakumar Varma (2007) and occasionally an earlier edition by Mangal Deva Shastri (1959).

³ For the purpose of this study, I am primarily using the edition of S. N. Sen and A. K. Bag (1983).

⁴ For editions of the Aṣṭādhyāyī used by me, see p. 145.

mean any existing linguistic or cultural practice established over a number of generations. It is something which one has received as the standard and would like to protect and pass on intact to the next generation, for example, the recitation of the Vedic *mantras*.

In order to achieve the goal of retention, these texts follow a common systematic approach which consists of two interdependent and complementary processes. The first one is an analysis of a given whole into constituent components and fundamental units. The other one is synthesis through rule-based combination of components and units to regain the given whole.

A basic operation which is needed for these processes is characterization of components and units. This involves attaching attributes to them to impart an identity and/or associating some information which they subsequently bear. The operations of characterization and combination are executed once certain conditions are satisfied.

The apparent cyclical exercise of first analyzing and then synthesizing comprehends the given phenomena in a systematic manner. This gives rise to an interconnected structure of components and units together with the conditioned operations. Such structures have the tendency to last longer and are explained on the basis of the underlying system. Structures facilitate variations and change.

In the following, I intend to demonstrate this systematic approach by means of examples to determine its chief features.

A.1 Analysis and synthesis

To begin with, we examine how the above mentioned texts perform the complementary processes of analysis and synthesis for the sake of comprehension of a given phenomenon.

A.1.1 Syllables (*Akṣaras*)

The first example we take up is the process of analyzing a given utterance into constituent syllables. Consider the following *mantra* of the Ṛgveda :

तत्सवितुर्वरेण्यम् । (RV. 3.62.10).

It is the given whole which can be analyzed into components. The components in this case are syllables (*akṣaras*). The Ṛgvedaprātiśākhya provides rules for analysis of a given utterance into constituting syllables. These are given as follows:

1. Both the short as well as long vowels (*svaras*) form a syllable.⁵
2. The nasal sound (*anusvāra*) and consonants (*vyañjana*) are part of the syllable.⁶
3. The consonants that are in between two vowels are part of the latter vowel.⁷
4. The nasal sound and aspirated sound (*visarjanīya*) form part of the preceding vowel.⁸
5. The first phoneme of a consonant conjunct (*saṃyoga*) if in between two vowels, optionally forms a part of the first vowel.⁹

An application of the above rules yields the following analysis. The separations are shown by the *danḍa* (|) sign.

तत् । स । वि । तुर् । व । रे । ण्यम् ।

It should be noted that the components are dependent upon the given whole and in this sense, do not have an independent existence. They have relevance only as long as they can be combined to form the original.

The results of analysis are not unique. This is because there are more than one possible options to dissect a given whole. Another variation of the above analysis could be:

त । त्स । वि । तु । र्व । रे । ण्यम् ।

The process of synthesis in this case is simple. It consists of only one rule, that of uttering or placing the analyzed components one after another, without any pause or space in between, while following the original sequential order. Both the processes of analysis and synthesis are rule-based. The rules of analysis are mentioned explicitly while the rules for synthesis are understood implicitly.¹⁰

⁵ उभये त्वक्षराणि । (RVPPr. 1.19).

⁶ अनुस्वारो व्यञ्जनं चाक्षराङ्गम् । (RVPPr. 1.22).

⁷ स्वरान्तरे व्यञ्जनान्युत्तरस्य । (RVPPr. 1.23).

⁸ पूर्वस्यानुस्वारविसर्जनीयौ । (RVPPr. 1.24).

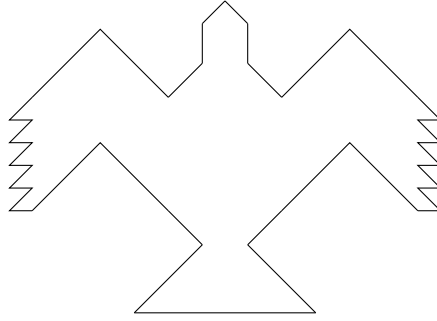
⁹ संयोगादिर्वा । (RVPPr. 1.25).

¹⁰ The rules to determine the syllables in the Vājasaneyiprātiśākhya are more detailed. Apart from the above mentioned cases, it also include instances when the consonants result from reduplication. For example, *pārśśvyam* (VS. 25.5) consists of *pārś* | *śvyam* where the first reduplicated consonant ś belongs to the former syllable and the second to the latter. See (VPr. 1.99-106).

A.1.2 The falcon shaped fire altar (*Śyenacit*)

We now consider an example from the Baudhāyanaśulbasūtra. The Śulbasūtras were primarily meant for piling-up of the fire altar (*agnicayana*) and preparation of the sacrificial arena.¹¹ Accordingly, these are usually mentioned as part of the general instructions for performance of the Vedic rituals (Śrautasūtras).¹² The Baudhāyanaśulbasūtra describes the construction of the sacrificial ground (*mahāvedi*), provides for the distance, relative positions and shape of the pits for the three fires—the *gārhapatya*, the *āhavanīya* and the *dakṣiṇāgni* as well as various altars (*vedis*), including piling-up of the bricks for fire altars (*citis*).

Consider the *śyenacit* or the fire altar in the shape of falcon with curved wings and extended tails.¹³



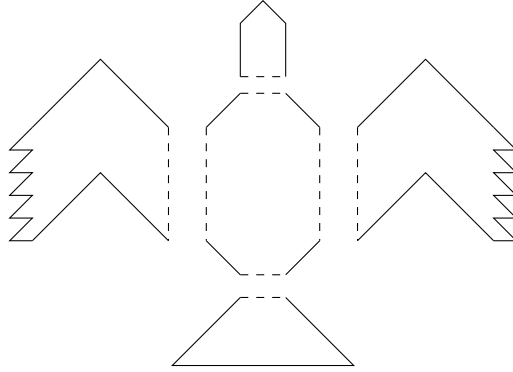
It is an altar having the above form and consisting of 1000 bricks piled in five layers, with 200 bricks in each layer. The entire area of the altar should be $7\frac{1}{2}$ times the area of a square with length equal to the height of the sacrificer (*yajamāna*).¹⁴ From a systematic point of view, this *śyenacit* altar is the given whole which is to be retained. The Śulba-sūtra enunciates its components. The figure is first divided in four parts: the main core (*ātman*), the head (*śiras*), the tail (*puccha*) and the two wings (*pakṣa*).

¹¹ On the terms *śulba* (or *śulva*) and Śulbasūtra see (Michaels 1978 p. 162-165). For an introduction to piling up of the fire altar (*agnicayana*), see (Michaels 1978 p. 36-43). Frits Staal (1983) provides a more detailed description of the ritual processes.

¹² There are Śulbasūtras ascribed to: Baudhāyana, Āpastamba, Vādhūla (belonging to the Taittirīyasamhitā of the Kṛṣṇayajurveda), Manu and Varāha (Maitrāyaṇīsamhitā of the Kṛṣṇayajurveda) as well as Kātyāyana (Vājasaneyīsamhitā of the Śuklayajurveda). Their relative as well as absolute chronology is unclear (Michaels 1978 p. 51-57). Scholars, however, place them between 5th to 2nd century BCE, with Baudhāyanaśulbasūtra being one of the earliest (Pingree 1981 p. 4-5).

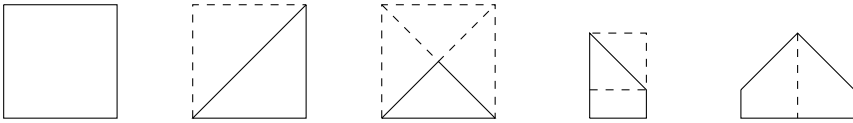
¹³ This is mentioned in the tenth chapter (BŚuS. 10.1-20). Here I follow the number according to the edition by S. N. Sen and A. K. Bag (1983).

¹⁴ See, for example, (BŚuS. 10.4-9).



The analysis is not unique. It could have been analyzed into a different set of components. Moreover, the analyzed components are dependent upon the given whole and lack independent existence or usage in the sacrificial activities.

The analyzed components are further dissected into five different shapes of smaller components that correspond to the five types of bricks.



1. The first type (B1) are square bricks having the size of one-fourth of a *puruṣa*. They are therefore called *caturthī* bricks.
2. The second type (B2) are triangular bricks obtained by cutting the *caturthī* bricks (B1) along the diagonal. These are called *ardhā* bricks as they have half the area of the B1 square bricks.
3. The third type (B3) are again triangular bricks obtained by cutting B1 across the diagonals. These are termed *pādyā* bricks having one-quarter of the area of B1.
4. The fourth type (B4) are four-sided quarter bricks formed by adding an isosceles triangle of two equal sides with length 1 pada and the hypotenuse with length $\sqrt{2}$ pada combined with a rectangle with along the length having measurements of length 1 pada and breadth $1\frac{1}{2}$ pada.
5. The fifth type (B5) are bricks obtained by joining two B4 bricks along their longest side. Its shape is like the beak of a swan, hence it is called *haṃsamukhī*.

The components in this case are evident and are literally the building blocks of the given whole. The components depend upon the original. They do not have independent employment or usage in the sacrificial rituals. Further, the

components are also not unique. There can be a different set of components, constituting the same whole.

The second set of components, namely the *caturthī*-bricks etc. are parts of bigger components like the main core (*ātman*) or the head (*śiras*) etc. Although more granular than the first set, the components belonging to the second set are formally speaking not different from the first.¹⁵

The process of synthesis is rule based. The nature of rules is placing the components in a manner so that the original form is regained. These are specified, for example, in (BŚuS. 10.10-20).

A.1.3 Saṃhitā-pāṭha and Pada-pāṭha

The main function of Prātiśākhya is to provide for the synthesis of continuous recitation (*saṃhitā-pāṭha*) from the word-for-word recitation (*pada-pāṭha*).¹⁶

An application of the process of analysis to the continuous recitations (*saṃhitā-pāṭha*)¹⁷ of the *mantras* of the Ṛgveda, yields individual *padas* or words. Their sequential enunciation is called word-for-word recitation (*pada-pāṭha*).¹⁸ The given whole in this case, therefore, is the continuous recitation of any *mantra* of the Ṛgveda and their analyzed elements i.e. the *padas* are its components.

The task of analyzing a given continuous expression into individual *padas* is largely a heuristic process and there are no rules following which

¹⁵ See below the discussion in section A.2.

¹⁶ The name Prātiśākhya indicates that they correspond to a particular branch (*śākhā*) of the Veda. Among the available ones, the Ṛgvedaprātiśākhya (attributed to Śaunaka), Taittirīyaprātiśākhya, Vājasaneyiprātiśākhya and Ṛktaṅtra are the most important. The chronology of these texts is not settled.

¹⁷ The definition of *saṃhitā* in Ṛgvedaprātiśākhya indicates the process of combining the components: पदान्तान्पदादिभिः संदधेति यत् सा कालव्यवायेन। (RVPr. 2.2). [*Saṃhitā* is that which combines the final parts of the *padas* with the initial parts of the following ones, without the intermission of time.] On the other hand, the Vājasaneyiprātiśākhya does not use the components *padas* but defines it on the basis of continuous recitation of the phonemes. वर्णानाम् एकप्राणयोगः संहिता। (VPr. 1.158). [*Saṃhitā* is the conjunction of phonemes uttered in one breath.]

¹⁸ The collections are respectively called the Saṃhitā-pāṭha (1933) and the Pada-pāṭha (1947). Vājasaneyiprātiśākhya terms the analyzed position as *asaṃhitā* in contradistinction to the combined state as *saṃhitā*. पूर्वणोत्तरः संहितः । पदविच्छेदोऽसंहितः । (VPr. 1.155-156). See (Varma 1987 p. 90-92).

the individual components can be identified.¹⁹ Their recognition is possible as they are meaningful components that are used interchangeably in the recitations.²⁰ It is facilitated by the process of concurrent presence (*anvaya*) and concurrent absence (*vyatireka*).²¹ As an example, we take the first *mantra* in the beginning of the Ṛgveda (1.1.1) and provide first the given whole and then its analyzed components:²²

Samhitā-pāṭha (the given whole)

agnim īle purohitam yajñasya devam ṛvijam / hotāram ratnadhātāmam /

Pada-pāṭha (analyzed components)

agnim / īle / puraḥ'hitam / yajñasya / devam / ṛvijam / hotāram / ratna'dhātāmam /

Prātiśākhya texts provide rules for combination of *padas* in order to regain the continuous recitation or Samhitā-pāṭha. The question as to what comes first—whether Pada-pāṭha has its origins in Samhitā-pāṭha or *vice versa*—is discussed by the commentators. The assertion संहिता पदप्रकृतिः in (ṚVPr. 2.1) can be understood in two different ways: *samhitā* whose *prakṛti* (origin) is *pada* or *samhitā*, which is the *prakṛti* (origin) of *pada*. Commenting on this, Uvaṭa says that:

samhitā, whose constituents are *padas*, is here a modification of the constituting *padas*. For example, the modifications *ṣatva* or *natva* occur in *samhitā* only. Because they are the constituents, therefore, *padas* are established original forms.²³

Yāska in his Nirukta also states that “*samhitā* is the one having *padas* as its constituent and all the branches of the Veda consider it to be so.”²⁴ Commenting

¹⁹ There are, however, some exceptions, for example, the fifth chapter of the Vājasaneyiprātiśākhya which gives some rules to identify the *padas* within the Samhitā-pāṭha. See (Varma 1987 p. 335-363).

²⁰ Vājasaneyiprātiśākhya defines a *pada* as that component, which is “capable of possessing an independent sense”. अर्थः पदम्। (VPr. 3.2). Yāska in Nirukta mentions four kinds of *padas* : nouns (*nāman*), verbs (*ākhyāta*), prepositions (*upasarga*) and indeclinables (*nipāta*). चत्वारि पदजातानि नामाख्याते चोपसर्गनिपाताश्च तानीमानि भवन्ति। (N. 1.1). K. V. Abhyankar provides a compendium on Pada-pāṭha and specifies *pada* (meaningful unit), their compact expression (*vṛtti*) and their types, *vigraha* or analysis of the conjoined expressions, resulting nodes or breaks (*parvan*), stops (*avagraha*), accentuation (*svara*) and other special features of Pada-pāṭha in (Abhyankar 1974).

²¹ On this process, see (Cardona 1967a p. 313-352).

²² The transliteration I use is the one generally adopted in western scholarship. I have not added accent marks here. George Cardona (1997 p. li-lxiv) discusses the different transliteration issues. Peter M. Scharf and Malcom D. Hyman (2012) provide a detailed study of various issues involved in encoding Sanskrit in computers.

²³ पदानि प्रकृतिभूतानि यस्याः संहितायाः सा पदप्रकृतिः संहितात्र विकारः। तथा हि षत्वणत्वादयो विकाराः संहिताया एव भवन्ति। प्रकृतिभूतत्वाच्च पदानां सिद्धत्वम्। Uvaṭa on (ṚVPr. 2.1).

²⁴ पदप्रकृतिः संहिता। पदप्रकृतीनि सर्वचरणानां पार्षदानि। (N. 1.17).

upon this, Durgācārya takes up the question in a detailed manner and puts forward two possible cases:

1. "That, which is the cause of *padas*, that (*saṃhitā*) is *pada-prakṛti*. Why? Because *padas* are formed out of *saṃhitā*. Therefore, some consider *saṃhitā* to be the original form (*prakṛti*) and *padas* to be their modifications (*vikāra*)."²⁵
2. "Others, however, understand the statement *padaprakṛtiḥ saṃhitā* to be *saṃhitā*, whose cause are the *padas*. Why? Because *saṃhitā* is gained out of the combinations of *padas* only. Therefore, *padas* are the original form and *saṃhitā* is their modification."²⁶

He further raises the question, which option is better: to consider *padas* to be the original form and the *saṃhitā* to be their modification or *vice versa* and decides for the latter giving several justifications based on the earlier usage of *saṃhitā-pāṭha*.²⁷ It follows that the Prātiśākhya texts perform the task of analysis of *saṃhitā-pāṭha* into *pada-pāṭha* and then provide a rule based process to combine the *padas* to regain the *saṃhitā-pāṭha*.

To sum up, we can identify the processes of analysis of a given whole into its components, and complementary to it the synthesis from components to the given whole in the examples taken from different texts. In particular we noticed the following instances:

1. Sanskrit expression into syllables (*akṣara*) : The example from the Prātiśākhya shows that both the process of analysis of a Sanskrit utterance into syllables as well as the reverse process of synthesis are rule based.
2. In case of a ritual formation like the *śyenacit* the process of analysis into components like the *caturthī* bricks is teleological. The reverse process of synthesis on the other hand is rule based.
3. The analysis of *saṃhitā-pāṭha* into the *pada-pāṭha* is again teleological, while synthesis is rule based.
4. The process of analysis of Sanskrit expressions into components like the roots (*prakṛti*) and suffixes (*pratyaya*) is again teleological, while synthesis is rule based.

We now look into the constituents that are gained by further analyzing the components.

²⁵ पदानां या प्रकृतिः सेयं पदप्रकृतिः । किं कारणम्? संहितातो हि पदानि प्रक्रियन्ते । तस्मात्संहितैव प्रकृतिर्विकारः पदानीत्येवमेके मन्यन्ते । Durgācārya on (N. 1.17).

²⁶ अपरे पुनः पदप्रकृतिः संहितेति पदानि प्रकृतिर्यस्याः सेयं पदप्रकृतिरिति । किं कारणम्? पदान्येव हि संहन्यमानानि संहिता भवति । तस्मात् पदान्येव हि प्रकृतिर्विकारः संहितेति । Durgācārya on (N. 1.17).

²⁷ आह । किं पुनरत्र साधीयः पदानां प्रकृतित्वं संहिताया विकारत्वमुत वा विकारत्वं पदानां प्रकृतित्वं संहिताया इति? उच्यते संहितायाः प्रकृतित्वं ज्यायः । आह । किं कारणम्? उच्यते । मन्त्रो ह्यभिव्यज्यमानः पूर्वमृषेर्मन्त्रदृशः संहितयैवाभिव्यज्यते न पदैः । Durgācārya on (N. 1.17).

A.2 Components and units

The process of analysis is carried further in the case of linguistic as well as geometrical components. The components are further sub-divided into fundamental units.

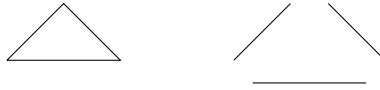
The linguistic components are analyzed into the basic sound units or the phonemes. For example,

Components: *tat / sa / vi / tur / va / re / nyam /*

Units: *t / a / t / s / a / v / i / t / u / r / v / a / r / e / ṅ / y / a / m /*

As in the case of components, the units are dependent upon the given phenomena. For example, the set of phonemes is dependent upon the type of sounds comprising a particular language. For some other language, this set may be different.²⁸ But, unlike the components, units are *unique* and *fundamental*. Uniqueness implies that there cannot be two different sets of units corresponding to the same phenomenon. And fundamental means that they can not be analyzed further.

The geometrical components are also further dissected into more basic units, namely points and lines. For example, the *pādyā* brick is analyzed as follows:



The set of units in this case would consist of *point*, *line*, *surface* etc. It should be noted, that as in the case of phonemes where, for example an element of the set represents a *type* that can have several instances, having different length and accents, similarly here, the elements stand for the fundamental *types* which may have different length or area etc. Thus, in this case as well, the set of units is fundamental and unique.

We have identified two different categories of constituents of a given phenomena.

1. The first category is what we name as components. Examples are *pada*, *prakṛti*, *pratyaya*, *akṣara*, *śīras*, *pucca*, *caturthī*-bricks.
2. The second category is what we define as the units. The examples of units are phonemes for linguistic-components and lines and points for geometrical-components.

²⁸ For example, certain sounds like *lh*, *jihvāmūliya*, *upadhmaniya* are a part of the Vedic expressions but not of classical Sanskrit.

To elucidate the difference between the components and units, we note that the units are fundamental constituents that cannot be divided further. What is meant here is that if one were to analyze them further, the same type would result. They are physical entities with some audible/visible form. In the case of linguistic units, the form would be the sound which one hears, and for geometrical units it is the visible shape. Further, we stipulate that a unit is characterized *only* through its form. In order to specify this, we need first to look into the process of characterization of units. Characterization in general is the process of identifying any object on the basis of certain features that help in differentiating it from dissimilar objects, and relating it with similar ones. The characteristics which identify units are related to their physical form. We call them features.

Let us look into the definitions in the Ṛgvedaprātiśākhya where the fundamental units are defined. These units, in the case of linguistic phenomena are the phonemes. The manner in which the characterizing features are assigned to them is by direct specification of the type: “the phoneme *x* has the feature *f*”. If more than one phoneme shares the same feature then they are referred to collectively. In order to refer to them, a list is provided at the beginning. The rules of the Ṛgvedaprātiśākhya refer to such a list. As an example, consider the very first rule: *aṣṭau samānākṣaraṇyāditaḥ* (RVPr. 1.1) [Eight are samānākṣara in the beginning]. This statement attaches the feature samānākṣara (monophthongs) to the phonemes: { a, ā, ṛ, ī, i, ī, u, ū }. Here, the relevant phonemes are put in a group. The name of the group is then the feature which is attached to the individual unit belonging to that set. In a similar manner, other features like *sandhyakṣara* (diphthongs), *svara* (vowels), *vyañjana* (consonants), *sparśa*, *varga* (the five groups of five each), *antaḥsthā* (semi-vowels), *ūṣman* (spirants), *aḥṣa* (sonant, voiceless), *soṣman* (aspirates), *kaṅṭhya* (velar) etc. are assigned to the respective phonemes.²⁹

Here, one must note the difference between *definition* of some sound and its *specification* within some linguistic expression. Consider the example of nasal sound (anunāsika). It is defined in that it is uttered in the list of sounds and later named or referred to as anunāsika.³⁰ In the Aṣṭādhyāyī, this defining characteristic about its phonetic form is mentioned as that utterance which is spoken simultaneously with mouth and nose.³¹ Its specification, on the other hand, is the determination that in a particular expression, some specific sound happens to be anunāsika.

²⁹ See (RVPr. 1.1-18).

³⁰ अनुनासिकोऽन्त्यः (RVPr. 1.14) [The last phoneme of each of the 5 groups is anunāsika].

³¹ मुखनासिकावचनोऽनुनासिकः ॥ १.१.८ ॥ ► from mouth and nose uttered sound is anunāsika.

A.2.1 Characterization

Let us now look at the characterization of components. A component is composed out of one or more units. In the case of linguistic components, this combination is a sequential concatenation of the individual sound units. Characterization of its physical/phonetic form, therefore, is in terms of the features of the constituent units. There are, however, other characteristics that are assigned to components. We call them attributes in order to distinguish them from the features. While features correspond only to those properties that refer to the physical form of a unit, attributes refer to the following additional aspects.

There are attributes whose assignment involves conditions which depend upon the distribution of phonemes within an expression. One such attribute is *guru* (heavy syllable). Apart from syllables having *dirgha* (long) vowel, the other ones, namely the one having *hrasva* (short) vowel, are also *guru* if *saṃyoga* (consonant cluster) or *anusvāra* (nasal sound) follows.³² There are attributes which depend upon the distribution of components, for example, the term *aṅga* which is attached to that part which is before the suffix for which some operation is specified.³³ The most common condition for attributes, however, is on the basis of their belonging to a particular set. For example, the group of suffixes (k)ta and (k)tavat(u) are called *niṣṭhā*.³⁴

The question that may be asked here is, why do we need these attributes? One requires attributes to formulate the rules of synthesis. It is very common that commentators mention the instances of rules where the attributes are used. For example, in the very first rule of the *Ṛgvedaprātiśākhya*, the purpose of coining the attribute *samānākṣara* is mentioned to be its use in formulation of a subsequent rule.³⁵ Thus, the main justification for specifying an attribute is its use in the formulation of the system. We would not need attributes if we do not want to formulate our rules in terms of them. Attributes, therefore, contain and represent grammatical information. They encode information which originates from the system.

³² गुरूणि दीर्घाणि। तथेतेषां संयोगानुस्वारपराणि यानि। (RVPr. 1.20-21).

³³ यस्मात्प्रत्ययविधिस्तदादि प्रत्ययेऽङ्गम् ॥ १.४.१३ ॥ ► that part which enjoins a pratyaya based operation, before that the sequence is *aṅga*.

³⁴ क्तवत् निष्ठा ॥ १.१.२६ ॥ ► (k)ta and (k)tavat(u) are *niṣṭhā*.

³⁵ समानाक्षरसंज्ञायाः प्रयोजनं समानाक्षरे सस्थाने इति। Uvaṭa on (RVPr. 1.1). The purpose of the term *samānākṣara* is its use in the rule *samānākṣare sस्थाने* (RVPr. 2.15).

A.2.2 Combination of units and components

In section A.1 we noted that the components are combined in a rule based manner which results in the synthesis of the original given whole. Thus, the syllables are combined to form the original linguistic expression. The *padas* of the Pada-pāṭha are combined following the rules of the corresponding Prātiśākhya to regain the original Saṃhitā-pāṭha. The bricks of the *śyenacit* like the *caturthī*, *ardhā*, *pādyā* etc. are combined to form the head (*śiras*), the main core (*ātman*) etc. which again are combined to form the original *śyenacit*. The components like roots (*prakṛti*) and suffixes (*pratyaya*) are combined to form expressions of standard speech.

A.2.3 Constructs

Next we consider combinations of units in order to form components. At this stage, we need to determine an important category of elements, which we term as the constructs. In order to introduce the basic idea about constructs, let us first consider an example from the Śulbasūtra. We have seen that a given phenomenon, like the *śyenacit*, is analyzed into components like head (*śiras*) and the main core (*ātman*), which are further dissected into components like the *caturthī*-bricks. Further, there are fundamental units, that are gained by analyzing the components. The question which can now be asked is: what can be constructed by combining the fundamental units in a rule based manner? As will soon become clear, we need to introduce a new category to represent the results of application of a rule-based procedure of combinations of the fundamental units. We call them constructs. For example, if we apply the rules mentioned in (BŚuS. 1.3-4) then we can form a square (*caturāśra*).

The difference between a component and a construct is that while the former is a tangible part of the given phenomenon, the latter is an abstraction of it resulting out of rule-based combinations of the units. The passage from the square-shaped *caturthī*-bricks to the concept of square, and consequently supplying rules for its formation is the significant step, which according to Axel Michaels (1978 p. 17-20), indicates the origin of the science of geometry. In our case, constructs represent not only the concept of a square, but equally important is the condition that they are *constructed* out of rule-based combinations of the fundamental units. Hence the choice of the term. It needs to be clarified here, that the constructs are also tangible and physical entities and represent the generalized concept of that entity and not its instantiation in the world of phenomena.

Another example is construction of syllables from the set of phonemes. The rules mentioned on page 132 specify this. Here again, syllables represent the conceptual constructs that are generated through rule-based combinations of phonemes. It should be noted that not all the syllables that can be theoretically constructed are instantiated as actual syllables belonging to some linguistic expression (*akṣaras*). Further, syllables as constructs are also physical entities. Syllables as components, however, are constituents of a given linguistic expression. A phoneme sequence like *bmha*, although a syllabic construct, is not a syllabic component of the Sanskrit language.³⁶

Constructs are abstractions of the components that are generated through a rule-based combination of units. Those constructs that are also constituents of a given phenomenon are instantiated as components of that phenomenon. This distinction between the constructs and components, as we shall see, is important to differentiate between the process of synthesis and the possibilities for formalization.

A.3 Structures, variations and change

The application of the systematic approach on a set of phenomena results in the evolution of a comprehending structure. The nature of such a structure is an interconnected network of components on the basis of their characteristics. The rules of synthesis specify this network or interconnection.

Thus, the rules of the *Prātiśākhya*s provide for the possible connections and combinations of the components (i.e. the *padas*) and the units (or the phonemes). Similarly, the rules of the *Śulbasūtras* specify the combinations of the components like the *caturthī* bricks as well as units or lines and points. In the case of grammar, again the rules provide for the interconnections of the components, i.e. which component can be combined with which other.

The systematic approach of analysis and synthesis not only provides a mechanism to retain a given phenomenon, but also facilitates the recording of variations and brings about change. There are two ways in which variation is comprehended. Firstly, while formulating the rules, provision for alternatives is provided. Pāṇini, for example, uses the expressions *anyatarasyām*, *vibhāṣā* and *vā* to accommodate the variations (Kiparsky 1980). Secondly, there are additional conditions which specify the varying contexts. In the case of the *Aṣṭādhyāyī*, some of these contextual variations are those corresponding to

³⁶ Ulrich Stiehl provides a list of attested syllabic components. See <http://www.sanskritweb.de/> (accessed on 09.07.2012).

the usage in the Vedic literature³⁷ or geographical³⁸ or even the opinions of others³⁹.

The existence of a system brings about consolidation of change by incorporating the *new* or not yet comprehended phenomena within the descriptive structure. This is facilitated because the new phenomena partially contains components that are already collected in the structure and partially extra fresh components are to be conceived. Changes can thus be integrated by minor adjustments in the structure.

³⁷ E.g. विभाषा छन्दसि ॥ १.२.३६ ॥ ► In recitation of Veda (*chandās*), ekaśruti is optional (*vibhāṣā*).

³⁸ E.g. एङ् प्राचां देशे ॥ १.१.७५ ॥ ► *eṅ* in case of expressions referring to the eastern region.

³⁹ E.g. संबुद्धौ शाकल्यस्येतावनार्षे ॥ १.१.१६ ॥ ► according to Śākalya o(t) is termed *pragrhya* if it is *sambuddhi* and when *iti* that is not Vedic (*ārṣa*) follows.

Bibliography

Sanskrit Texts

Aṣṭādhyāyī

- Rama Nath Sharma (6 vols.) 1987, 1990, 1995, 1999, 2001 and 2003.
- Sumitra M. Katre 1989.
- S. D. Joshi and J. A. F. Roodbergen (14 vols.)

Baudhāyanaśulbasūtra

- S. N. Sen and A. K. Bag 1983.

Kāśikāvṛtti

- Jaya Shankar Lal Tripathi (5 vols.) 1986a, 1986b, 1986c, 1987, 1988.

Mahābhāṣya

- Franz Kielhorn (3 vols.) 1885.
- S. D. Joshi and J. A. F. Roodbergen 1975.

Nirukta

- Laksman Sarup (2 vols.) 1962, 1967.

Paribhāṣenduśekhara

- Charudev Shastri 1973.

Paspāśāhnika

- S. D. Joshi and J. A. F. Roodbergen 1986.

Ṛgveda

- *Ṛgvedasaṃhitā: Śrīmatsāyanācāryaviracitabhāṣyasametā.* (5 vols.) 1933-51, Vaidika Saṃśodhana Maṇḍala, Pune.

Ṛgvedaprātiśākhya

- Virendra Kumar Varma 2007.

Siddhāntakaumudī

- S. C. Vasu and V. D. Vasu (3 vols.) 1905, 1906, 1907.

Vājasaneyiprātiśākhya

- Virendra Kumar Varma 1987.

Secondary Texts

Abhyankar, Kashinath Vasudev (1974). *Veda-padapāṭha-caracā*. Bhandarkar Oriental Research Institute.

Abhyankar, Kashinath Vasudev and Shukla, Jaydev Mohanlal (1977). *A dictionary of Sanskrit grammar*. Oriental Institute, Baroda.

Bharati, Akshar and Chaitanya, Vineet and Sangal, Rajeev (1994). *Natural Language Processing: A Paninian Perspective*. Prentice-Hall of India, New Delhi.

Bare, James Stanton (1980). *Phonetics and Phonology in Pāṇini: The System of Features Implicit in the Aṣṭādhyāyī*. Department of Linguistics, The University of Michigan, Ann Arbor, Michigan.

Bhate, Saroja (1987). The meaning-adhikāras in the taddhita section of the Aṣṭādhyāyī: An analysis. In: *Indo-Iranian Journal* 30 (2), p. 81-92. Brill.

Bhate, Saroja and Kak, Subhash (1993). Pāṇini's Grammar and Computer Science. In: *Annals of the Bhandarkar Oriental Research Institute* 72, p. 79-94.

Bhattacharya, R. S. (1955). Kinds of Expositions in Sanskrit Literature. In: *Annals of the Bhandarkar Oriental Research Institute* 36, p. 123-132.

— — — (1966). *Pāṇinīya Vyākaraṇa kā Anuśīlana*. Indological Book House, Varanasi.

Bloomfield, Leonard (1929). Review of Konkordanz Pāṇini-Candra by Bruno Liebich. In: *Language* 5 (4), p. 267-276. Linguistic Society of America.

Breloer, B. (1929). Studie zu Panini. In: *Zeitschrift für Indologie und Iranistik* 7, p. 114-135.

— — — (1935-36). Die 14 pratyahara-sutras des Panini. In: *Zeitschrift für Indologie und Iranistik* 10, p. 133-191.

Briggs, Rick (1985). Knowledge Representation in Sanskrit and Artificial Intelligence. In: *AI Magazine* 6 (1), p. 32-39.

— — — (1987). Conference Report: Knowledge Representation and Inference in Sanskrit. In: *AI Magazine* 8 (2), p. 99.

Bronkhorst, Johannes (1979). The role of meanings in Pāṇini's grammar. In: *Indian Linguistics* 40, p. 146-157.

Cardona, George (1965). On Pāṇini's Morphophonemic Principles. In: *Language* 41 (2), p. 225-237.

— — — (1967). Pāṇini's syntactic categories. In: *Journal of the Oriental Institute* 16 (3), p. 201-215.

— — — (1969). Studies in Indian Grammarians. I. The Method of Description Reflected in the Śivasūtras. In: *Transactions of the American Philosophical Society* 59 (1), p. 3-48. American Philosophical Society.

— — — (1974). Pāṇini's Kāraḥ: Agency, Animation and Identity. In: *Journal of Indian Philosophy* 2, p. 231-306.

— — — (1980). *Pāṇini. A Survey of Research*. Motilal Banarsidass, Delhi.

— — — (1997). *Pāṇini: His Work and its Traditions. Volume One: Background and Introduction*. Motilal Banarsidass, Delhi.

— — — (1999). *Recent research in Pāṇinian studies*. Motilal Banarsidass, Delhi.

— — — (2009). On the structure of Pāṇini's System. In: *Sanskrit Computational Linguistics Symposium*. Huet, Gérard and Kulkarni, Amba and Scharf, Peter Ed(s). p. 1-32. Springer, Heidelberg.

Chatterji, K. C. (1964). *Technical terms and technique of Sanskrit grammar*. Calcutta University Press, Calcutta.

Chattopadhyay, Suhrid Sankar and Chaudhuri, Kalyan (2001). An event in Kolkata. In: *Frontline* 18-25, p. 15-19.

Chomsky, Noam (1956). Three models for the description of language. In: *IRE Transactions on Information Theory* 2 (3), p. 113-124.

— — — (1957). *Syntactic structures*. The Hague: Mouton.

— — — (1959). On certain formal properties of grammars. In: *Information and Control* 2, p. 137-167.

— — — (1963). Introduction to the formal analysis of natural languages. *Handbook of Mathematical Psychology*. Vol. 2, Luce, R. Duncan and Bush, Robert R. and Galanter, Eugene Ed(s). p. 269-322. John Wiley and Sons.

Chomsky, Noam and Shützenberger, Marcel P. (1963). The algebraic theory of context free languages. In: *Computer Programming and Formal Languages*. Braffort, P. and Hirschberg, D. Ed(s). p. 118-161. North-Holland.

Clark, John and Yallop, Colin and Fletcher, Janet (2007). *An introduction to phonetics and phonology*. Blackwell, Massachusetts.

Deo, Ashwini (2003). Derivational Morphology in Inheritance-based Lexica: Insights from Pāṇini. Paper read at the 12th World Sanskrit Conference, held in Helsinki in 2003.

DeRose, Steven J. (1997). *The SGML FAQ Book: Understanding the Foundation of HTML and XML*. Kluwer Academic, Boston.

Deshpande, Madhav M. (1985). Historical Change and the Theology of Eternal Sanskrit. In: *Zeitschrift für vergleichende Sprachforschung* 98, p. 122-149.

— — — (1990). Semantics of Kāraṅkas in Pāṇini: An Exploration of Philosophical and Linguistical Issues. In: *Sanskrit and Related Studies: Contemporary Researches and Reflections*. Matilal, B.K. and Bilimoria, Purusottama Ed(s). p. 33-57. Sri Satguru Publications, Delhi.

— — — (1992). Pāṇini in the Context of Modernity. In: *Language and Text-Studies in Honour of Ashok R. Kelkar*. Srivastava, R. N. Ed(s). p. 15-27. Kalinga Publications, Delhi.

Devasthali, G. V. (1967). *Anubandhas of Pāṇini*. University of Poona, Poona.

Dīkṣita, Puṣpā (2006). *Aṣṭādhyāyī saḥajabodha : Kṛdantaprakaraṇam: (samasta dhātuṃ meṃ sāre kṛt pratyaya lagāne kī apūrova vidhi)*. Pratibhā Prakāśana, Dillī.

— — — (2006). *Aṣṭādhyāyī saḥajabodha : Taddhitakaraṇam: (samasta dhātuṃ meṃ sāre kṛt pratyaya lagāne kī apūrova vidhi)*. Pratibhā Prakāśana, Dillī.

— — — (2007). *Aṣṭādhyāyī sahajabodha : Sārvadhātuka khaṇḍa*. Pratibhā Prakāśana, Dillī.

— — — (2007). *Aṣṭādhyāyī sahajabodha : Ārdhadhātuka khaṇḍa*. Pratibhā Prakāśana, Dillī.

Dvivedi, Hari Prapanna (1978). *Studies in Pāṇini: Technical terms of the Aṣṭādhyāyī*. Inter-India Publications, Delhi.

Fowler, Murray (1965). How Ordered Are Pāṇini's Rules?. In: *Journal of the American Oriental Society* 85 (1), p. 44-47. American Oriental Society.

Gonda, Jan (1975). *Vedic literature: (Saṃhitās and Brāhmanas)*. Harras-sowitz, Wiesbaden.

Goyal, Pawan and Kulkarni, Amba and Behera, Laxmidhar (2009). Computer Simulation of Aṣṭādhyāyī: Some Insights. In: *Sanskrit Computational Linguistics Symposium*. Huet, Gérard and Kulkarni, Amba and Scharf, Peter Ed(s). p. 139-161. Springer, Heidelberg.

Grimal, François Ed(s). (2005). *Pāṇinītyavyākaraṇodāharaṇakośaḥ*. Rashtriya Sanskrit Vidyapitha and École Française d'Extrême-Orient, Tirupati/Paris.

— — — (2006). *Udāharaṇasamāhāraḥ*. Rashtriya Sanskrit Vidyapitha, Tirupati.

— — — (2006a). *Samāsaprakaraṇam*. Rashtriya Sanskrit Vidyapitha, Tirupati.

— — — (2010). *Tiṅantaparakaraṇam*. Rashtriya Sanskrit Vidyapitha, Tirupati.

Haag, Pascale and Vergiani, Vincenzo Ed(s). (2009). *Studies in the Kāśikāvṛtti: the section on Pratyāhāras; critical edition, translation and other contributions*. Società Editrice Fiorentina, Firenze.

Hellwig, Oliver (2009). SanskritTagger, a stochastic lexical and POS tagger for Sanskrit. In: *Sanskrit Computational Linguistics Symposium*. Huet, Gérard and Kulkarni, Amba and Scharf, Peter Ed(s). p. 266-277. Springer, Heidelberg.

— — — (2010). Performance of a lexical and POS tagger for Sanskrit. In: *Sanskrit Computational Linguistics-4th International Symposium*. Jha, Girish Nath Ed(s). p. 162-172. Springer.

Houben, Jan E. M. (1999). 'Meaning statements' in Pāṇini's grammar: on the purpose and context of the Aṣṭādhyāyī. In: *Studien zur Indologie und Iranistik* 22, p. 23-54.

— — — (2008). Bhaṭṭoji Dīkṣit's "Small Step" for a Grammarian and "Giant Leap" for Sanskrit Grammar. In: *Journal of Indian Philosophy* 36, p. 563-574.

— — — (2009). Pāṇini's Grammar and Its Computerization: A Construction Grammar Approach. In: *Sanskrit Computational Linguistics Symposium*. Kulkarni, Amba and Huet, Gérard Ed(s). p. 6-25. Springer, Heidelberg.

Hueckstedt, Robert A. (1995). *Nearness and Respective Correlation: A History of the Interpretations of Aṣṭādhyāyī 6.1.77: iko yaṅ aci*. Harrassowitz Verlag, Wiesbaden.

Huet, Gérard and Kulkarni, Amba and Scharf, Peter Ed(s). (2009). *Sanskrit Computational Linguistics-First and Second International Symposia*. Springer, Heidelberg.

Hyman, Malcolm D. (2009). From Pāṇinian Sandhi to Finite State Calculus. In: *Sanskrit Computational Linguistics Symposium*. Huet, Gérard and Kulkarni, Amba and Scharf, Peter Ed(s). p. 253-265. Springer, Heidelberg.

Jha, Girish Nath Ed(s). (2010). *Sanskrit Computational Linguistics-4th International Symposium*. Springer, Heidelberg.

Joshi, S. D. (1968). *Patañjali's Vyākaraṇa-Mahābhāshya Samarthāhnikā (P.2.1.1): Edited with Translation and Explanatory Notes*. University of Poona, Pune.

— — — (2001). Syntactic and Semantic Devices in the Aṣṭādhyāyī of Pāṇini. In: *Journal of Indian Philosophy* 29, p. 155-167.

— — — (2009). Background of the Aṣṭādhyāyī. In: *Sanskrit Computational Linguistics Symposium*. Kulkarni, Amba and Huet, Gérard Ed(s). p. 1-5. Springer, Heidelberg.

Joshi, Shivram Dattatray and Bhate, Saroja (1984). *The fundamentals of Anuṣṭi*. University of Poona, Pune.

Joshi, S. D. and Roodbergen, J. A. F. (1975). *Patañjali's Vyākaraṇa Mahābhāshya Kārikāhnikā (P.1.4.23-1.4.55): Introduction, Translation and Notes*. University of Poona, Pune.

— — — (1980). *Patañjali's Vyākaraṇa Mahābhāshya Vibhaktiyāhnikā P.2.3.18-2.3.45: Introduction, Text, Translation and Notes*. University of Poona, Pune.

— — — (1983). The Structure of the Aṣṭādhyāyī. In: *Proceedings of the International Seminar on Studies in the Aṣṭādhyāyī of Pāṇini*. Joshi, S. D. and Laddu, S. D. Ed(s). University of Poona, Pune.

— — — (1986). *Patañjali's Vyākaraṇa Mahābhāṣya Paspasāhnikā: Introduction, Text, Translation and Notes*. University of Poona, Pune.

— — — (1991). *The Aṣṭādhyāyī of Pāṇini: With Translation and Explanatory Notes. Vol. I*. Sahitya Akademi, New Delhi.

— — — (1994). *The Aṣṭādhyāyī of Pāṇini: With Translation and Explanatory Notes. Vol. III*. Sahitya Akademi, New Delhi.

Katre, Sumitra M. (1989). *Aṣṭādhyāyī of Pāṇini*. Motilal Banarsidass, Delhi.

Kiparsky, Paul (1982). *Some theoretical problems in Pāṇini's grammar*. Bhandarkar Oriental Research Institute, Poona.

— — — (1991). On Pāṇinian Studies: A Reply to Cardona. In: *Journal of Indian Philosophy* 19, p. 331-367.

— — — (1991a). Economy and the Construction of the Śivasūtras. In: *Pāṇinian Studies. Professor S.D. Joshi Felicitation Volume*. Deshpande, Madhav M. and Bhate, Saroja Ed(s). University of Michigan: Center for South and Southeast Asian Studies.

— — — (2002). On the Architecture of Pāṇini's Grammar. CIEFL Conference on the Architecture of Grammar (15.-17.1.2002).

— — — (2009). On the Architecture of Pāṇini's Grammar. In: *Sanskrit Computational Linguistics Symposium*. Huet, Gérard and Kulkarni, Amba and Scharf, Peter Ed(s). p. 33-94. Springer, Heidelberg.

Kiparsky, Paul and Staal, J. F. (1969). Syntactic and Semantic Relations in Pāṇini. In: *Foundations of Language* 5 (1), p. 83-117. Springer.

Kulkarni, Amba and Huet, Gérard Ed(s). (2009). *Sanskrit Computational Linguistics-Third International Symposium*. Springer, Heidelberg.

Misra, Vidya Niwas (1964). Pāṇini's Grammar as a mathematical model. In: *Indian Linguistics* 24, p. 157-178.

— — — (1966). *The Descriptive Technique of Pāṇini: An Introduction*. Mouton & Co., The Hague.

Mylius, Klaus (1980). Über die Anwendung mathematischer Methoden in der Vedaforschung. In: *Ethnographisch-Archäologische Zeitschrift* 21, p. 233-248.

Nagarajao, H. V. (1978). The Scope and Necessity of Aṅgādhikāra. In: *Journal of Indian Philosophy* 6, p. 145-176.

Pandit, M. D. (1966). *Mathematical Representation of some Pāṇinian Sūtras*. University of Poona, Pune.

— — — (1974). Formal and non-formal in Pāṇini. In: *Annals of the Bhandarkar Oriental Research Institute* 54, p. 179-192.

Petersen, Wiebke (2004). A Mathematical Analysis of Pāṇini's Śivasūtras. In: *Journal of Logic, Language and Information* 13, p. 471-489. Kluwer Academic Publishers, Netherland.

— — — (2008). Zur Minimalität von Pāṇinis Śivasūtras-Eine Untersuchung mit Methoden der Formalen Begriffsanalyse. University of Düsseldorf.

— — — (2009). On the Construction of Śivasūtra-Alphabets. In: *Sanskrit Computational Linguistics Symposium*. Kulkarni, Amba and Huet, Gérard Ed(s). p. 79-98. Springer, Heidelberg.

Popper, Karl Raimund (1959 / 2005). *The logic of scientific discovery*. Routledge, London.

Rau, Wilhelm (2002). *Bhartṛharis Vākyapadīya : Versuch einer vollständigen deutschen Übersetzung nach der kritischen Edition der Mūla-Kārikās*. Akademie der Wissenschaften und der Literatur, Mainz.

Renou, Louis (1963). Sur le genre Sūtra dans la littérature sanskrite. In: *Journal Asiatique* 251, p. 165-216.

Sarup, Lakshman (1962). *The Nighaṅṭu and The Nirukta: Introduction, English Translation and Notes*. Motilal Banarsidass, Delhi.

— — — (1967). *The Nighaṅṭu and The Nirukta: Sanskrit Text with an Appendix Showing the Relation of the Nirukta with other Sanskrit Works*. Motilal Banarsidass, Delhi.

Scharf, Peter M. (2009). Modeling Pāṇinian Grammar. In: *Sanskrit Computational Linguistics*. Gérard Huet, Amba Kulkarni and Scharf, Peter Ed(s). p. 95-126. Springer, Heidelberg.

Scharfe, Hartmut (1971). *Pāṇini's Metalanguage*. American Philosophical Society, Philadelphia.

— — — (2009). A new perspective on Pāṇini. In: *Indologica Taurinensia* vol. XII, p. 1-273.

Sen, Samarendra Nath and Bag, Amulya Kumar (1983). *The Śulbasūtras of Baudhāyana, Āpastamba, Kātyāyana and Mānava: with text, English transl. and comm. by S. N. Sen and A. K. Bag.* Indian National Science Academy, New Delhi.

Sharma, Rama Nath (1975). Referential Indices in Pāṇini. In: *Indo-Iranian Journal* 17 (1-2), p. 31-39. Brill.

— — — (1987). *The Aṣṭādhyāyī of Pāṇini (Volume I): Introduction to the Aṣṭādhyāyī as a Grammatical Device.* Munshiram Manoharlal.

— — — (1990). *The Aṣṭādhyāyī of Pāṇini: (Volume II) English translation of Adhyāya one with Sanskrit text, translation, word-boundary, Anuvṛtti, Vṛtti, explanatory notes, derivational history of examples, and indices.* Munshiram Manoharlal, New Delhi.

— — — (1995). *The Aṣṭādhyāyī of Pāṇini: (Volume III).* Munshiram Manoharlal, New Delhi.

— — — (1999). *The Aṣṭādhyāyī of Pāṇini: (Volume IV).* Munshiram Manoharlal, New Delhi.

— — — (2001). *The Aṣṭādhyāyī of Pāṇini: (Volume V).* Munshiram Manoharlal, New Delhi.

— — — (2003). *The Aṣṭādhyāyī of Pāṇini: (Volume VI).* Munshiram Manoharlal, New Delhi.

Śāstrī, Cārudeva (1970). *Vyākaraṇacandrodaya : Kṛt va taddhita.* Motilal Banarsidass.

— — — (1971). *Vyākaraṇacandrodaya : Tiṅanta.* Motilal Banarsidass.

— — — (1972). *Vyākaraṇacandrodaya : Strīpratyaya, subanta, avyaya.* Motilal Banarsidass.

— — — (1973). *Śikṣā, sañjñā, paribhāṣā, saṃhitā, liṅga.*

— — — (1987). *Vyākaraṇacandrodaya : Kāraka va samāsa.* Motilal Banarsidass.

Singh, J. D. (1979). Technical Terms in Pāṇini. In: *International Journal of Dravidian Linguistics* 8, p. 7-16.

Staal, J. F. (1962). A Method of Linguistic Description: The Order of Consonants According to Pāṇini. In: *Language* 38 (1), p. 1-10. Linguistic Society of America.

— — — (1965). Context-Sensitive Rules in Pāṇini. In: *Foundations of Language* 1 (1), p. 63-72. Springer.

— — — (1966). Pāṇini Tested by Fowler's Automaton. In: *Journal of the American Oriental Society* 86 (2), p. 206-209. American Oriental Society.

— — — (1975). The concept of metalanguage and its Indian background. In: *Journal of Indian Philosophy* 3, p. 315-354.

— — — (1992). Sūtra. In: *Kalātattvakośa: a lexicon of fundamental concepts of the Indian arts*. Vatsyayan, Kapila Ed(s). p. 303-314. Indira Gandhi National Centre for the Arts, New Delhi.

Subbanna, Sridhar and Varakhedi, Srinivasa (2009). Computational Structure of the Aṣṭādhyāyī and Conflict Resolution Techniques. In: *Sanskrit Computational Linguistics Symposium*. Kulkarni, Amba and Huet, Gérard Ed(s). p. 56-65. Springer, Heidelberg.

Subrahmanyam, Prakya Sreesaila (1999). *Paṇinian Linguistics*. Institute for the Study of Languages and Cultures of Asia and Africa, Tokyo University of Foreign Studies, Tokyo.

Thieme, Paul (1956). Pāṇini and the Pāṇinīyas. In: *Journal of the American Oriental Society* 76 (1), p. 1-23. American Oriental Society.

Tripathi, Jaya Shankar Lal and Malaviya, Sudhakar Ed(s). (1986). *Kāśikā of Vāmana and Jayāditya (Adhyāya 1.1-1.2) with Nyāsa of Jinendrabuddhi, Padamañjarī of Haradatta and Bhāvabodhinī Hindi Commentary by J.S.L. Tripathi*. Tara Printing Works, Varanasi.

— — — (1986). (*Adhyāya 1.3-2.1*). Tara Printing Works, Varanasi.

— — — (1986). (*Adhyāya 2.3-3.2*). Tara Printing Works, Varanasi.

— — — (1987). (*Adhyāya 3.3-4.1*). Tara Printing Works, Varanasi.

— — — (1988). (*Adhyāya 4.2-5.1*). Tara Printing Works, Varanasi.

Nooten, Barend A. (1969). Pāṇini's Theory of Verbal Meaning. In: *Foundations of Language* 5, p. 242-255.

Varma, Virendra Kumar *Ed(s)*. (1987). *Vājasaneyi-Prātisākhya*. Chaukhamba Sanskrit Pratishthan, Delhi.

— — — (2007). *Śaunaka viracitaṃ R̥gveda-prātisākhyaṃ (Uvaṭa bhāṣya saṃvalitaṃ)*. Caukhambā Saṃskṛta Pratiṣṭhāna, Dillī.

Vasu, S. C. and Vasu, V. D. *Ed(s)*. (1905). *Siddhānta Kaumudī of Bhaṭṭojī Dīkṣita (Volume I)*. Panini Office, Bhuvaneśvara Āśrama, Allahabad.

— — — (1906). (*Volume II*): *Verbs*. Panini Office, Bhuvaneśvara Āśrama, Allahabad.

— — — (1907). (*Volume III*). Panini Office, Bhuvaneśvara Āśrama, Allahabad.

Wezler, Albrecht (2001). Zu der Frage des "Strebens nach äußerster Kürze" in den Śrautasūtras. In: *Zeitschrift der Deutschen Morgenländischen Gesellschaft* 151, Harrassowitz Verlag.

Whitney, William Dwight (1885). *The Roots, Verb-Forms and Primary Derivatives of the Sanskrit Language*. Breitkopf and Härtel, Leipzig.

Wilson, Edgar Bright (1952). *An introduction to scientific research*. McGraw-Hill, New York.

Wujastyk, Dominik (1993). *Metarules of Pāṇinian grammar: Vyāḍi's Paribhāṣāvṛtti*. Forsten, Groningen.

The present work is a study of the Aṣṭādhyāyī of Pāṇini from a new perspective. It attempts to explore the Pāṇinian system of Sanskrit grammar from a formal point of view and investigate the possibilities of representing it in a logical, explicit and consistent manner. It puts forward an appropriate framework for such a representation. Differing from the formulation of Aṣṭādhyāyī, which is composed in an artificial yet natural language and is meant to be employed by individuals who are acquainted both with the Sanskrit language and the techniques of grammar, the present rendering aims for a non-verbal representation in terms of mathematical categories and logical relations which can be implemented in an algorithmic manner. The formal framework suggested in this work would facilitate adequate tools for postulating and evaluating hypotheses about the grammatical system. Moreover, it would furnish the basis for a computer implementation of the grammar. Both these aspects are objects of enquiry in the field of theoretical studies on Pāṇini as well as the emerging discipline of Sanskrit computational linguistics. This book takes on the ground-work in these areas.



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

ISBN 978-3-946054-55-9



9 783946 054559