Informatics in Pāṇini’s Grammar

Amba Kulkarni

Department of Sanskrit Studies,
University of Hyderabad,
Hyderabad

International Seminar on History and Philosophy of Mathematics, Kolkata

February 21, 2014
Outline

Structure of Aṣṭādhyāyī

Analysis of Sanskrit Language
The mathematical method is characteristic of much of Western philosophy whereas the grammatical method is characteristic of much of Indian philosophy.

Informatics

Etymology: Informatics
  German: Informatik
  French: Informatique
Informatics

Etymology: Informatics

German: Informatik
French: Informatique

Information + automatique $\Rightarrow$ informatique
Informatics: Science of Automatic processing of Information
Information coding device: Language
Informatics

Etymology: Informatics

German: Informatik
French: Informatique

Information + automatique \Rightarrow informatique
Informatics: Science of Automatic processing of Information
Information coding device: Language

Computational Linguistics \subset Informatics
Informatics

**Etymology:** Informatics

- German: Informatik
- French: Informatique

**Information + automatique ⇒ informatique**

Informatics: Science of Automatic processing of Information

Information coding device: Language

Computational Linguistics ⊂ Informatics

**The First Formal system of Grammar: Pāṇini’s Aṣṭādhyāyī**
Pāṇini’s Aṣṭādhyāyī

Circa 500 B.C.E.
Extant Grammar of the then prevalent Sanskrit Language
Around 4000 sutras$^1$(aphorism)
8 chapters 4 sections each

$^1$alpāksaram(concise) asandigdham(unambiguous) sāravat(contains essence)
viśvatomukham(general) |
astobham(continuous) anavadyam(without flaw) ca sūtraḥ sūtravido viduḥ||(vāyu
purāṇa)
It is admired for its
  simplicity
completeness of the descriptive coverage
rigorous consistency in the use of meta language
intricate system of conventions governing rule application and rule interaction
richness in various aspects of informatics
One of the greatest monuments of human intelligence (Bloomfield) is only beginning to claim its rightful position in linguistics. Many of the insights of Panini’s grammar still remain to be recaptured, but those that are already understood constitute a major theoretical contribution.

Paul Kiparsky, Emeritus Prof. Stanford University
Panini, then, was not an ancient and nebulous precursor of a science in which everything has since been done better, but a distant colleague of genius from whom linguists are still able to learn.

Prof. Fritz Staal

Not only Panini was by far the first linguist in recorded history, but I claim he was the first informaticien, 24 centuries before computers came into existence.

From informatics point of view, the importance of Aṣṭādhyāyī is two fold,

The Structure of Aṣṭādhyāyī
Method of Analysis of Sanskrit Language
Aṣṭādhyāyī consists of around 4000 aphorisms with some ancillary texts.

śivasūtras (special order of the phonemes)
dhātupāṭha (list of verbal roots)
gaṇapāṭha (various sets of nouns)
liṅgānuṣāsanam (system for deciding the gender)
uṇādi sūtras (?) (special rules)
Normal Arrangement of Alphabet

| Vowels   | a  ā i ī u ũ ō ṓ ā́́| e ai o au aṃ aḥ |
|-----------|------------------|
| Velar     | ka kha ga gha ŏa |
| Palatal   | ca cha ja jha ŏa |
| Retroflex | ŏa ŏha ŏa ŏha ŏa |
| Dental    | ta tha ḍa ḍha ŏa |
| Labiel    | pa pha ba bha ma |
| semi-vowel| ya ra la va |
| Fricative | śa ṣa sa ha |
Pāṇini required several (42) subsets of this alphabet to describe various operations.
śivasūtrāṇi

Some of these subsets:

All vowels
All consonants
All vowels + semivowels + nasals
The voiceless stops
stops and sibilants
Any of ba ga ḍa da
Any of bha gha ḍha dha
Any of ya va ra ṇa ma ṇa ṇa na ma jha bha
....
It is not advisable to give 42 names to these sets. It will be difficult to memorize the association.

These are Partially ordered sets.

Pāṇini arranged them linearly in the form of 14 ShivasUtras.
śivasūtrāṇi

1. a i u (N)
2. r l (K)
3. e o (N)
4. ai au (C)
5. ha ya va ra (Ṭ)
6. la (N)
7. ā ma ā na na (M)
8. jha bha(Ñ)
9. gha ḍha dha (Ś)
10. ja ba ga ḍa da (Ś)
11. kha pha cha ṭha tha ca ṭa ta (V)
12. ka pa (Y)
13. śa śa s (R)
14. ha (L)
Justification of this arrangement is attempted independently by

Cardona (on historical grounds)
Stall (linguistically)
Kiparsky (logically)
and Petersen (mathematically)

Petersen (2008) proved that the arrangement is optimal.
And is one among the 12 000 000 possibilities.
śivasūtrāṇi

The set *khar*

1. a i u (N)
2. r l (K)
3. e o (Ñ)
4. ai au (C)
5. ha ya va ra (Ñ)
6. la (N)
7. ŋa ma ŋa ŋa na (M)
8. jha bha (Ñ)
9. gha ḍha dha (Ś)
10. ja ba ga ḍa da (Ś)
11. kha pha cha ṭha tha ca ṭa ta (V)
12. ka pa (Y)
13. śa śa s (R)
14. ha (L)
śivasūtrāṇī

1. a i u (N)
2. {r l} (K)
3. {e o} (Ñ)
4. {ai au} (C)
5. ha ya va ra (T)
6. la (N)
7. ŋa ma {ña ŋa na} (M)
8. jha bha (Ñ)
9. {gha ḍha dha} (S)
10. ja {ba ga ḍa da} (Ś)
11. {kha pha} {cha ṭha thā} {ca ṭa ta} (V)
12. {ka pa} (Y)
13. {śa śa s} (R)
14. ha (L)

2! * 2! * 2! * 2! * 3! * 3! * 4! * 2! * 3! * 3! * 2! * 3! ≈ 12 000 000
Given a set of Partially Ordered Sets,
Now it is possible to tell
Whether the elements are
Śivasūtra encodable or not.
Weibke Petersen (2008)
Structure of Aṣṭādhyāyī

A) The First Formal Grammar

a) A Formal Grammar is written in a Formal Language.
b) The Formal language has well-defined Syntax.

Aṣṭādhyāyī is written in Sanskrit.
The syntax as well as the programme are intermixed in the same piece of work.
Formal Grammar

\[ G = (N, \Sigma, P, S) \]

- \( N \): A finite set of Non-terminals
- \( S \): The Start Symbol \( \in N \)
- \( \Sigma \): A finite set of Terminals
- \( P \): Production rules, of the type
  \[ (\Sigma \cup N)^* N (\Sigma \cup N)^* \rightarrow (\Sigma \cup N)^* \]
Example 1

\[
\text{padam (word)} ::= \text{subantam (nominal form)}
\]

\[
\text{subantam (nominal form)} ::= \text{ti\text{n}antam (verbal form)}
\]

\[
\text{subantam (nominal form)} ::= \text{prätipadikam (nom stem) sup (nom suff)}
\]

\[
\text{prätipadikam (nominal stem)} ::= \text{kr̥t (noun derived from a verb)}
\]

\[
\text{prätipadikam (nominal stem)} ::= \text{taddhita (noun derived from a noun)}
\]

\[
\text{prätipadikam (nominal stem)} ::= \text{samāsa (compound)}
\]

\[
\text{samāsa (compound)} ::= \text{underived prätipadikam (nominal stem)}
\]

\[
\text{alaukika vigrahaḥ (Intermediate ling exp)}
\]

\[
\text{alaukika vigrahaḥ} ::= \text{prätipadikam sup prätipadikam sup (saha supā 2.1.4)}
\]
Syntax of Āṣṭādhyāyī

\[
A ::= B \\
    | C \\
    ;
B ::= D \ E
D ::= F \\
    | G \\
    | H \\
    | I \\
    ;
H ::= J
J ::= K \ '+' \ L \ M \ '++' \ N \\
    ;
\]
Syntax of Aṣṭādhyāyī

Example 2:

iko yanaci (6.1.77)

ikk{6} yan{1} ac{7}
Syntax of Aṣṭādhyāyī

Example 2:
iko yaṇaci (6.1.77)

ik{6} yaṇ{1} ac{7}

tasmin iti nirdiṣṭe pūrvasya (1.1.65)
A word ending in locative case indicates ‘of the preceding’.

ṣaṣṭhī sthāneyogā (1.1.48)
A word in genitive case undergoes change.
Syntax of Aṣṭādhyāyī

Example 2:
iko yaṇaci (6.1.77)

ik{6} yaṇ{1} ac{7}

tasmin iti nirdiśṭe pūrvasya (1.1.65)
A word ending in locative case indicates ‘of the preceeding’.

ṣaṣṭhī sthāneyogā (1.1.48)
A word in genitive case undergoes change.

ik{6} ac{7} ::= yaṇ{1} ac{7}
Example 2:

iko yaṇaci (6.1.77)

\[
\text{iko}\{6\} \text{ yaṇ}\{1\} \text{ ac}\{7\}
\]

tasmin iti nirdiśte pūrvasya (1.1.65)
A word ending in locative case indicates ‘of the preceeding’.

\[
\text{ṣaṣṭhī}\ \text{sthāneyogā} \ (1.1.48)
\]
A word in genitive case undergoes change.

\[
\text{ik}\{6\} \ \text{ac}\{7\} ::= \text{yaṇ}\{1\} \ \text{ac}\{7\}
\]
\[
\{i,u,r,l\} \ \{a,i,u,r,l,e,o,ai,au\} \rightarrow \{y,v,r,l\} \ \{a,i,u,r,l,e,o,ai,au\}
\]
Syntax of Aṣṭādhyāyī

tasmin iti nirdiśṭe pūrvasya (1.1.65)
A word ending in locative case indicates ‘of the preceeding’.

tasmāt iti uttarasya (1.1.66)
A word ending in ablative case indicates ‘of the following’.

ṣaṣṭhi sthāneyogā (1.1.48)
A word in genitive case undergoes change.

\[
W\{5\} \ W\{6\}W\{7\} ::= W\{5\} \ W\{1\}W\{7\}
\]

Ingermann observed that the sūtras of Aṣṭādhyāyī have the same structure as that of BNF and suggested to rename ‘Backus Naur Form’ as ‘Pāṇini Backus Naur Form’ (1967 ACM Communications).
Distributivity in Mathematics:
\[ a \times (b + c) = a \times b + a \times c \]

Distributivity in Languages:

*John goes home.*
*John eats fruits.*

*John goes home and eats fruits*

Or simply,

*John goes home,*
*eats fruits.*
Consider the following sūtras:

\textit{upadeśe ac anunāsik it} 1.3.2
\textit{hal anyam} 1.3.3
\textit{na vibhaktau tuskāḥ} 1.3.4
\textit{ādiḥ niṭuḍavāḥ} 1.3.5
\textit{ṣaḥ pratyayasya} 1.3.6
\textit{cuṭū} 1.3.7
\textit{laṣaku ataddhite} 1.3.8
Anuvṛtti

upadeśe it

ac anunāsik (=it)
hal antyam

na vibhaktau tumāḥ (=it)

ādiḥ

niṭuḍavāḥ (=it)
pratyayasya

śaḥ (=it)
cuṭū (=it)
laśaku (=it) ataddhite
Anuvṛtti

upadeśe (a) it (c)

ac anunāsik ( = it)(b)
hal antyam (d)

na vibhaktau tvsmāḥ(=it) (e)

ādiḥ (f)

ñiṭuḍavāḥ (=it) (g)
pratyayasya (h)

ṣaḥ(=it) (i)
cuṭū (=it) (j)
laśaku (=it) ataddhite (k)
Anuvṛtti

a (b + de + f [ g + h { i + j + k } ] ) c
No Proper Nesting; *maṇḍūka plutīḥ*

1.1.1 *vṛddhiḥ ādaic*

1.1.2 *adeṅ guṇah*

1.1.3 *ikāḥ guṇavrddhī (vṛddhiḥ guṇah)*
Maximum advantage of features of Natural Language:

How are the complete phrases reconstructed?

ākāṅkṣāḥ (Expectancy): Major role in deciding the anuvṛtti (Bhate)
Anuvṛttiḥ

Example of borrowing from as many as 12 sūtras

Original sūtra: 3-3-65 kvaṇaḥ vīṇāyām ca
After Anuvṛtti: pratyayaḥ paraḥ ca (ādyudāttah ca dhātoḥ kṛt
kriyārthāyām bhāve akartari ca kārake sajñāyām) ap upasarge vā nau
Anuvṛttiḥ

Some Statistics:

- Total Sūtras: (3,984) 4,000
- Total Words: (7,007) 7,000
- Total Words after repeating the words with anuvṛtti: 40,000
- Compression due to anuvṛtti: \(\frac{40,000}{7,000} \approx \frac{1}{6}\)
- In terms of byte size, compression: \(\frac{1}{3}\)

Significant from Oral Tradition.

The time to memorise the sūtras grows exponentially.

With anuvṛtti, a student can memorise the complete Aṣṭādhyāyī in about only 6 months!
Ordering of the rules

\[ pūrvatra \text{ asiddham} \ (8.2.1) \]
\[ \text{asiddhavat atra ābhāt} \ (6.4.22) \]
\[ \text{ṣatvatukorasiddhaḥ} \ (6.1.83) \]
Ordering of the rules

asiddhavat atra ābhāt (6.4.22)

hujhulbhyoh herdhiḥ (6.4.101)
śāhau (6.435)

6.4.101: śās + hi → śās dhi
6.4.35: śās + hi → śā + hi

Application of one rule blocks the application of the other. Both the rules need to be applied.

asiddhavat atra ābhāt (6.4.22) → Parallel Computing
Conflict Resolution

Programming Languages
  Longest match
  Precedence Rules

Aṣṭādhyāyī

vipratiṣedhe param kāryam (1.4.2)
Modularity

The rules related to a particular task are grouped together. For example, consider the following sūtras which identify sounds used as markers (anubandha).

\[ upadeśe ac anunāsik it 1.3.2 \]
\[ hal antyam 1.3.3 \]
\[ na vibhaktau tusmāḥ 1.3.4 \]
\[ ādiḥ niṣṭavāḥ 1.3.5 \]
\[ śaḥ pratyayasya 1.3.6 \]
\[ cuṭū 1.3.7 \]
\[ laṣaku ataddhite 1.3.8 \]
Features of Aṣṭādhyāyī .. contd

Object Oriented Programming:

Encapsulation of data with the (markers to the) functions
Bhaj + (gh)a(ṅ): In the presence of gh, j → g

Inheritance:

Multiple inheritance → arranged as a linear inheritance
derivational suffixes, deriving a noun from a noun (Taddhita pratyaya)
(Ashwini Deo 2007)
Pāṇini paid utmost attention to the dynamics of Information flow while analysing Sanskrit.

We cite 3 sūtras to highlight this point.

- anabhihite(1.3.1) *(Where is the information Coded)*
- svatantraḥ kartā (1.4.54) *(How much information is coded)*
- samānakartṛkayoḥ pūrvakāle (3.4.21) *(How is the information coded)*
anabhihite (Where is the information coded?)

Skt: rāmaḥ rāvaṇam hanti.
Gloss: Rama{nom} Ravana{acc} kill{3p sg pres}
Eng: Rama killed Ravana.

Skt: rāmeṇa rāvaṇaḥ hataḥ.
Gloss: Rama{inst} Ravana{nom} kill{3p sg past pl}
Eng: Ravana was killed by Rama.
svatantraḥ kartā (1.4.54) (*How much* information is coded)

John opened the door with a key.
This key opened the door.
The wind opened the door.
The door opened.
svatantraḥ kartā (1.4.54) *(How much information is coded)*

John opened the door with a key.
This key opened the door.
The wind opened the door.
The door opened.

John opened the door with a key. Agent
This key opened the door. Instrument
The wind opened the door. Cause
The door opened. goal

In Pāṇinian Framework, all of them are kartā!
Greatness of *Pāṇini* lies in identifying **EXACTLY HOW MUCH** information is coded in a language string. 
⇒ Upper Bound for the possible Analysis using only a language string and grammar.
We can extract precisely only that information which can be decoded from the language string ‘without any requirement of additional knowledge’.

Analogy:
We can not do high quality work with low quality energy.
Dhanyavādaḥ
Thank you for your attention