Abstract

Pāṇini in his Asṭādhyāyī has written the grammar of Sanskrit in an extremely concise manner in the form of about 4000 sūtras. We have attempted to mathematically remodel the data produced by these sūtras. The mathematical modelling is a way to show that the Pāṇinian approach is a minimal method of capturing the grammatical data for Sanskrit which is a natural language. The sūtras written by Pāṇini can be written as functions, that is for a single input the function produces a single output of the form y=f(x), where x and y is the input and output respectively. However, we observe that for some input dhātus, we get multiple outputs. For such cases, we have written multivalued functions that is the functions which give two or more outputs for a single input. In other words, multivalued function is a way to represent optional output forms which is expressed in Pāṇinian grammar with the help of 3 terms i.e. vā, vibhaṣā, and anyatarasyām. Comparison between the techniques employed by Pāṇini and our notation of functions helps us understand how Pāṇinian techniques ensure brevity and terseness, hence illustrating that Pāṇinian grammar is minimal.

1 Introduction

Pāṇini’s Asṭādhyāyī is ‘almost an exhaustive grammar for any human language with meticulous details yet small enough to memorize it’ (Kulkarni, 2016). Such an exhaustive grammar is ideal to be used for artificial language processing. Briggs (Briggs, 1985) even demonstrated in his article the salient feature of Sanskrit language that can make it serve as an artificial language. Although, this is not a new concept, various efforts in mathematical modelling of Indian languages have been done before. Joseph Kallrath in his book ‘Modeling Languages in Mathematical Optimization’ says that ‘a modeling language serves the need to pass data and a mathematical model description to a solver in the same way that people especially mathematicians describe those problems to each other’ (Kallrath, 2013). Mathematical modelling of languages also impacts our understanding of the language and its grammar. As scholars are delving into the question of formalizing various natural languages, it is also having an impact on how we understand the language itself. Recent work in theoretical and computational linguistics has influenced the interpretation of grammar (Scharf, 2008). We have followed a similar approach, wherein we have modelled the Pratyayas in Sanskrit in the form of functions with the help of Pāṇinian sūtras.

Similar to mathematical functions which can be expressed as f(x)=y where x is the input and y is the output of function f; the sūtras too look for their preconditions in an input environment. The effects produced by sūtras become part of an ever-evolving environment which may trigger other’ (Sohoni & Kulkarni, 2018). For the grammar to fit mathematical functions, we ‘need a strong and unambiguous grammar which is provided by Maharishi Pāṇini in the form of Asṭādhyāyī’ (Agrawal, 2013).

Statistical analysis of a language is a vital part of natural language processing (Goyal, 2011). According to how components of the target linguistic phenomenon are realized.
mathematically, available models of language evolution can be classified as rule-based and equation-based models. Equation-based models tend to transform linguistic and relevant behaviors into mathematical equations (Tao Gong, 2013), which is what we have attempted in this paper.

Ambiguity is inherent in the Natural Language sentences (Tapaswi & Jain, 2012), and hence Sanskrit being a natural language also has certain ambiguities. The ambiguity that we are dealing with in this paper is that a single dhātu combined with a single pratayaya can result in two or more optional forms. Mathematical modelling of such natural languages can help to remove this ambiguity. Traditionally too, there have been attempts by various scholars like Kātyāyana, Patanjali and Bhartrhari to provide extensive commentaries which contain explanations for various aspects of the grammar. They do not question Pāṇini’s basic model, but rather explain it, refine it and complete it (Huet, 2003). Explanations and clarifications in the form of various vārtikas also come handy while dealing with ambiguities. However, here we are diverging from the traditional approach and writing functions in order to model the grammatical data.

To account for more than two forms of a word, Pāṇini uses optional form rules to state that alternate forms are also possible. For example, śūtra (rule) 1.2.3 vibhaṣorṇoḥ states that ‘After the verb ṭṛṇa ‘to cover’, the affix beginning with the augment it is regarded optionally like īt (Source, 2020).’ We have used multivalued functions to denote such optional forms in our system of representing the pratayayas as functions.

2 Methodology

We are here attempting to mathematically model the data produced by the śūtras for which we started with compiling the list of dhātus and their respective derived dhātus with different pratayayas like from the Kridantkosh of Pushpa Dikshita Vol.I (Dikshita, 2014), sanskritworld.in (Dhaval Patel, n.d.), Siddhananta Kaumudi of Bhattoji Dikshita (S.C.Vasu, 1905), The Madhaviya DhātuVritti (Sayanacarya, 1964) and the roots, verb-forms and primary derivatives of the Sanskrit Language by W.D.Whitney (Whitney, 1885). The list of dhātus without the application of any pratayaya are considered as x, after the application of the concept of anubandhas. Anubandhas have a very prominent role to play in the Pāṇinian system of Sanskrit grammar. It literally means ‘what is attached to’. It has been used by all ancient authorities on Sanskrit grammar who have come after Pāṇini, right from Kātyāyana to Nageśa. However, Pāṇini has used the term ‘it’ to describe the anubandhas. M. Williams dictionary (Williams, 2008 revised) defines anubandhas as an indicative letter or syllable attached to roots etc., marking some peculiarity in their inflection e.g. an ‘i’ attached to roots denotes the insertion of a nasal before their final consonant. According to Nyāyakosa, anubandha is a letter that is attached to the stem (prakṛti), termination (pratyaya), augment (āgama) or a substitute (ādesha) to indicate the occurrence of some special modifications such as vikaraṇa, āgama, gūṇa or vṛddhi, accent etc. But it is dropped from the finished word i.e. pada. The use of anubandha is one of the crucial steps Pāṇini has taken to ensure the brevity and terseness of his work. We can say that anubandhas do form part of the pratayayas etc. to which they are found appended (Devasthali, 1967). But before we directly start writing our functions, we need to define the input set which comprises of dhātus from the Dhātupatha as well as the derived dhātus without anubandhas.

Let A be a set of all the dhātus after the anubandhas have been removed. These primary dhātus are 1943 in total. However, the input dhātus are not limited to these dhātus in set A. We can also derive a new dhātu set B by adding a san pratayaya to the dhātus of set A. The items in set B can be called dhātus by following the grammatical rule laid down by Pāṇini, ‘3.1.32 sanādyantāḥ dhātavah’ which says that ‘all roots ending with

<table>
<thead>
<tr>
<th>Śūtra numbers</th>
<th>Pratayaya</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.5</td>
<td>san</td>
</tr>
<tr>
<td>3.1.8</td>
<td>kyac</td>
</tr>
<tr>
<td>3.1.9</td>
<td>kāmyac</td>
</tr>
<tr>
<td>3.1.11</td>
<td>kyaḥ</td>
</tr>
<tr>
<td>3.1.13</td>
<td>kyaṣ</td>
</tr>
<tr>
<td>3.1.20</td>
<td>niḥ</td>
</tr>
<tr>
<td>3.1.21</td>
<td>nic</td>
</tr>
<tr>
<td>3.1.22</td>
<td>yaḥ</td>
</tr>
<tr>
<td>3.1.27</td>
<td>yak</td>
</tr>
<tr>
<td>3.1.28</td>
<td>āy</td>
</tr>
<tr>
<td>3.1.29</td>
<td>īyaḥ</td>
</tr>
</tbody>
</table>

Table 1: List of San pratayayas in Aṣṭādhyāyī with their respective śūtra numbers
the *pratyayas* starting with san are called *dhātu*.

Hence the input $x$ is defined as,

$$x \in (A \cup B)$$

In this paper we will focus on the multivalued functions that give two or more outputs for the same input *dhātu* of the form $f(x) = \begin{cases} y_1 \\ y_2 \\ y_3 \end{cases}$ if there are two optional forms; $f(x) = \begin{cases} y_1 \\ y_2 \\ y_3 \end{cases}$ if there are three optional forms and so on.

### 3 Notation

Let $x$ be the input *dhātu*. For the purpose of writing these functions, we start enumerating the syllables from left to right or from right to left depending upon that particular function. We can denote $x$ as, $x = (\ldots, x(2), x(1)) = (x'(1), x'(2), \ldots)$. $x$ can be a consonant (C) or a vowel (V) and they are denoted by

- $C'(i) = i^{th}$ consonant from left;
- $V'(i) = i^{th}$ vowel from left;
- $C(i) = i^{th}$ consonant from right,
- $V(i) = i^{th}$ vowel from right.

<table>
<thead>
<tr>
<th>cura =</th>
<th>c</th>
<th>u</th>
<th>r</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right to left</td>
<td>x(4)</td>
<td>x(3)</td>
<td>x(2)</td>
<td>x(1)</td>
</tr>
<tr>
<td>Left to right</td>
<td>x'(1)</td>
<td>x'(2)</td>
<td>x'(3)</td>
<td>x'(4)</td>
</tr>
</tbody>
</table>

| Table 2: The numbers 1, 2, 3, … signify the position of the syllable. The notation $x$ (unprimed) is used when the syllables are counted right to left, and the notation $x'$ is used when the syllables are counted left to right. For example: If $x = cura$, then Conversion are denoted by a right arrow with a number on the top. The number denotes the location of the conversion.

For example, $x[a \rightarrow \bar{a}]$ denotes that in the *dhātu* $x$, $a$ which is at the 2nd place from the right is getting replaced with $\bar{a}$.

We also define a ‘+ operator’ to explain the change of syllables when two syllables combine. In Sanskrit language when two syllables come closer, for the ease of pronunciation (in most cases) it gets replaced by another syllable or a combination of syllables. For example: $u+i=vi$, $e+i=ayi$, $o+i=avi$, $d+ta=ta$, $ch+t=ṣṭa$, $j+ta=kta$, $dh+ta=dhda$, $bh+ta=bdha$, $h+ta=nḍha$. Note that although the ‘+ operator’ may look similar to the concept of *Sandhi* in Sanskrit, it is totally based on our need to fit our dataset and does not encompass the broad concept of *Sandhi*.

### 4 A function $p(x)$

This function is not a *pratyaya* function, but it is required to write the *pratyaya* function. Thus, it would be helpful to define it here. The *dhātus* which have two or more vowels are called *udātta*, and when a suffix is added to them an additional ‘i’ comes. Such *dhātus* are called *seṭ* (literally meaning ‘with it’). For *dhātus* which have one vowel, we need to see the instructions given in the *Dhātupatha*. They can either be seṭ or aniṭ depending upon the given instructions given. Example of one such instruction is ‘*bhu sattayāṃ| udāṭṭḥ parasmaibhāṣḥ*’ It says that ‘i’ will come as the prayogsamavāyī svara is *udātth*.

The function $p(x)$ is defined by,

$$p(x) = \begin{cases} 1 & \text{if } x \text{ is set}, \\
0 & \text{if } x \text{ is aniṭ}. \end{cases}$$

Figure 1: Example of an Instruction given in the *Dhātupatha*.

$p(x) = \begin{cases} 1 & \text{if } x \text{ is set}, \\
0 & \text{if } x \text{ is aniṭ}. \end{cases}$
5 Multivalued functions

The words used for optionality by Pāṇini are vā, vibhaṣā, anyatarasyām. vā appears 136 times, vibhaṣā appears 258 times and, anyatarasyām appears 161 times respectively in Aṣṭādhyāyī; including the ones that occur in Anuvritti1. Pāṇini and all the commentators have given us no indication that they are supposed to be anything but synonyms. But the modern scholar Paul Kiparsky has wondered how could this be so, because Pāṇini has vowed to eliminate every needless extraneous syllable and there must be a deeper reason to suggest the use of three different terms. Hence, he has propounded the hypothesis in his well-argued study Pāṇini as a ‘variationist’ that the three terms vā, vibhaṣā, anyatarasyām refer respectively to three different kinds of options: those that are preferable (vā), those that are marginal (vibhaṣā) and those that are simple options(anyatarasyām) (Sharma, 2018).

One such case which results in such optional forms is represented in the table below where the addition and absence of ‘i’ results in two forms and the change of ‘h’ syllable to two different syllables further results in two forms. Thus, we end up with three forms of the same word. Let us look at an example for this case for x = muh:

\[
tum(x) = \begin{cases} 
    x[i \rightarrow e o] + 0 + tum \\
    x[i \rightarrow e o] + 0 + dhum \\
    x[i \rightarrow e o] + i + tum \\
\end{cases}
\]

<table>
<thead>
<tr>
<th>Word</th>
<th>Occurrence</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>vā</td>
<td>136 times</td>
<td>preferable</td>
</tr>
<tr>
<td>vibhaṣā</td>
<td>258 times</td>
<td>marginal</td>
</tr>
<tr>
<td>anyatarasyām</td>
<td>161 times</td>
<td>simple options</td>
</tr>
</tbody>
</table>

Table 3: Words used for optionality by Pāṇini

are marginal (vibhaṣā) and those that are simple options(anyatarasyām) (Sharma, 2018).

Some Multivalued functions for Tumun

Case I:
If \( x \in \{svr sū dhū\} \), then

\[
tum(x) = \begin{cases} 
    x[i/iu/ũr/i \rightarrow e o ar] + i + tum \\
    x[i/iu/ũr/i \rightarrow e o ar] + 0 + tum \\
\end{cases}
\]

<table>
<thead>
<tr>
<th>( x )</th>
<th>( tum(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>sū</td>
<td>svi</td>
</tr>
<tr>
<td>sūvi</td>
<td>svitum</td>
</tr>
<tr>
<td>sūv</td>
<td>svar</td>
</tr>
<tr>
<td>svaritum</td>
<td>svaritum</td>
</tr>
<tr>
<td>sūtum</td>
<td>svaritum</td>
</tr>
</tbody>
</table>

Case II:
If \( x \) has two syllables such that \( x(1) = ō \), then

\[
tum(x) = \begin{cases} 
    x[ũ \rightarrow ar] + i + tum \\
    x[ũ \rightarrow ar] + 0 + tum \\
\end{cases}
\]

<table>
<thead>
<tr>
<th>( x )</th>
<th>( tum(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ōv</td>
<td>varītum</td>
</tr>
<tr>
<td>varī</td>
<td>varītum</td>
</tr>
<tr>
<td>kō</td>
<td>karītum</td>
</tr>
<tr>
<td>karī</td>
<td>karītum</td>
</tr>
</tbody>
</table>

Case III:
If \( x \in \{gup\} \), then

\[
tum(x) = \begin{cases} 
    x[u \rightarrow o] + i + tum \\
    x[u \rightarrow o] + 0 + tum \\
    x[u \rightarrow o] + ā + i + tum \\
\end{cases}
\]

6 Cases for multivalued functions

Some cases for multivalued functions are displayed below2.

Some Multivalued functions for Tumun
Pratyaya

Case I:
If \( x \in \{svr sū dhū\} \), then

\[
tum(x) = \begin{cases} 
    x[i/iu/ũr/i \rightarrow e o ar] + i + tum \\
    x[i/iu/ũr/i \rightarrow e o ar] + 0 + tum \\
\end{cases}
\]

<table>
<thead>
<tr>
<th>( x )</th>
<th>( tum(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>sū</td>
<td>svi</td>
</tr>
<tr>
<td>sūvi</td>
<td>svitum</td>
</tr>
<tr>
<td>sūv</td>
<td>svar</td>
</tr>
<tr>
<td>svaritum</td>
<td>svaritum</td>
</tr>
<tr>
<td>sūtum</td>
<td>svaritum</td>
</tr>
</tbody>
</table>

Figure 2: Multivalued functions

---

1 The number of times these words appear in Aṣṭādhyāyī;

2 An exhaustive list of cases for Tumun and san pratyayas including the multivalued cases are given in the appendix in Devanagari script.
Case IV:
If \( x \in \{ t \tilde{r} p d \tilde{r} p \} \), then
\[
\text{tum}(x) = \begin{cases} 
\frac{x}{r} + 0 + \text{tum} \\
\frac{x}{r} + 0 + \text{tum} \\
\frac{x}{r} + i + \text{tum}
\end{cases}
\]

gup  |  gopi |  gopitum
-----|-------|-------
gop  |  gop   |  gopitum
gopāyi  |  gopitum |  gopāyitum

Case I:
If \( x'(1)=c, x'(2)=v= i u, x'(3)=c \) in \( x \) (which has exactly 3 letters), then
\[
\text{san}(x)=\begin{cases} 
T(x) + v'(1) + x[i u \rightarrow o a] + i\tilde{a} & \\
T(x) + v'(1) + x + i\tilde{a}
\end{cases}
\]
where, \( T(x)=c'(1)[k g h s h \rightarrow c j b s j] \)

Case II:
If there is only one \( v \) in \( x \), such that \( x(2)=v= i u \) and starts with at least two consonants i.e \( x'(1)=c, x'(2)=c \), then
\[
\text{san}(x)=\begin{cases} 
c'(1) + v'(1) + x[v[i u \rightarrow o a]] + p(x) + \tilde{a} & \\
c'(1) + v'(1) + x + p(x) + \tilde{a}
\end{cases}
\]

Some Multivalued functions for San Pratyaya

<table>
<thead>
<tr>
<th>( x )</th>
<th>c'(1) + v'(1)</th>
<th>san(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cyut</td>
<td>cu</td>
<td>cuciotisa</td>
</tr>
<tr>
<td>kliš</td>
<td>ci</td>
<td>cicleśa</td>
</tr>
</tbody>
</table>

7 Conclusion

According to the mathematical definition of a function, it generates a unique output for every input. However, while mathematically modelling Pratyayas in Sanskrit we came across several instances where a single input was generating multiple outputs, which have been represented by multivalued functions.

To ensure brevity, Pāṇini has used several tools which have been compared with their equivalent tools in our functional approach.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Pāṇinian tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>x(2)</td>
<td>upadhā</td>
</tr>
<tr>
<td>c'1,c'2,..,v'1 if x'=consonant; c'1,c'2,..,v'2 if x'=vowel</td>
<td>ekāc</td>
</tr>
<tr>
<td>x(1)</td>
<td>antya</td>
</tr>
<tr>
<td>-</td>
<td>anuvṛtti</td>
</tr>
</tbody>
</table>

Table 1: Pāṇinian Techniques vs functions

What we are essentially denoting as \( x(2) \) in our functions i.e. the penultimate term is nothing but upadhā. Pāṇini by convention treats \( x(1) \) as the end and calls it antya. This is clear from the definition of upadhā given by Pāṇini in Aṣṭādhyāyī sūtra ‘1.1.65 alontyāt pūrva upadhā’, which means ‘The letter immediately preceding the last letter of a word is called penultimate (upadhā) (Creative Commons, 2020)’. As stated before in the paper, the words vā, vibhaṣā, and anyatarasyam are used by Pāṇini to denote optional forms that we have denoted by multivalued functions.

Another important feature of Pāṇinian grammar is anuvṛtti, which is a technique of carrying some parts of the previous sūtras to the next sūtras. Due to anuvṛtti, the order in which various elements appear in the sūtra itself are very important. However, we do not need to define any such equivalent tool in our modeling as long as
we define some global functions and operators such as \( p(x) \) and the \( '+' \) operator.

By mathematically modeling *pratyayas*, the reason behind use of these techniques employed by *Pāṇini* to ensure brevity becomes very clear.

Mathematical modelling of *Pāṇinian* grammar in this way helps identify some general patterns, each of which is grouped separately as a case in the functions. These patterns are mainly dependent upon the occurrence of certain specific syllables at certain places. However, we observed that there are some *dhātus* which even after fulfilling the conditions given in the cases, give an output which is different from what is observed in the literature. All such cases needed a separate approach. Hence the for the treatment of such cases input sets for those particular cases have been defined.

The knowledge of *Pāṇinian* rules also helps us reduce the number of individual cases that have been constructed for each function. It helps group certain cases together into a single generalized case. For example: instead of writing three individual functions for \( i \rightarrow e, u \rightarrow o, \) and \( r \rightarrow ar \), the knowledge of the rules in *Aṣṭādhyāyī* helps to write a general case of the form \( i u r \rightarrow e o ar \).

Writing such functions for all other *pratyaya* functions may lead us towards a global function for *pratyayas* and for other grammatical tools as well. This technique of mathematical modelling is extremely helpful to understand Sanskrit grammar for people who are non-linguists or do not understand the technicalities of Sanskrit grammar. This mathematical model can also form a base for further processing of the grammatical rules for natural language processing of the language with the help of well-defined input and output sets.

**Acknowledgments**

Our deepest regards to Prof Swapneel Mahajan from the Department of Mathematics, IIT Bombay whose guidance in terms of inputs and ideas have helped shape the concept of these functions.

**References**


