Computational Algorithms Based on the Paninian System to Process Euphonic Conjunctions for Word Searches

Kasmir Raja S. V.
Dean – Research
SRM University
Chennai, India

Rajitha V.
Department of Computer Science
Meenakshi College for Women
Chennai, India

Meenakshi Lakshmanan
Department of Computer Science
Meenakshi College for Women
Chennai, India
meenakshi.lakshmanan@gmail.com

Abstract – Searching for words in Sanskrit E-text is a problem that is accompanied by complexities introduced by features of Sanskrit such as euphonic conjunctions or ‘sandhis’. A word could occur in an E-text in a transformed form owing to the operation of rules of sandhi. Simple word search would not yield these transformed forms of the word. Further, there is no search engine in the literature that can comprehensively search for words in Sanskrit E-texts taking euphonic conjunctions into account. This work presents an optimal binary representational schema for letters of the Sanskrit alphabet along with algorithms to efficiently process the sandhi rules of Sanskrit grammar. The work further presents an algorithm that uses the sandhi processing algorithm to perform a comprehensive word search on E-text.

Keywords – Sanskrit; euphonic conjunction; sandhi; linguistics; Panini; Sanskrit word search; E-text search.

I. INTRODUCTION

Word search in Sanskrit E-texts is a problem that is beset with complexities, unlike in the case of English E-texts. The problem assumes relevance in the context of the availability of rapidly increasing numbers of ancient Sanskrit texts [5–9] in the electronic format. The importance of n-gram analysis of Sanskrit texts for scholars and the tremendous utility of locating specific words in a variety of texts to aid the scholastic process can hardly be overemphasized.

Dating of a text, fixing its authorship with certainty, and analysis of the writing style of an author of a text, are some of the areas in which n-grams assume criticality especially in the context of ancient Sanskrit works. Quoting from authoritative texts is imperative in scholarly works, and word searches can provide crucial help in this regard. Locating the portion in a text or texts in which a particular usage or word is found is of great importance to scholars who write explanatory treatises of Sanskrit-based works in English and other languages. Semantic analysis and understanding of texts are facilitated by finding occurrences of words and studying them in different contexts. In fact, ancient Sanskrit works are universally acknowledged as being mines of information on a whole spectrum of disciplines, and hence finding actual occurrences of words is of great consequence to not only Sanskrit scholars but to also researchers from various other disciplines ranging from philosophy, theology, the arts and the physical and life sciences to sociology, medicine and astronomy.

II. THE PROBLEM

As stated above, there are complexities involved in searching comprehensively for words in a Sanskrit E-text. One of the major contributors to this complexity is the operation of euphonic conjunctions or ‘sandhis’. A sandhi is a point in a word or between words, at which adjacent letters coalesce and transform [3]. This is a common feature in many Indian languages as against European languages, and has far-reaching consequences in Sanskrit. The transformation caused by the application of rules of sandhi in Sanskrit can be significant enough to alter the word itself to such a degree that the transformed word would not show up in a simple word search.

For example, the word ‘āsamarddhiḥ’ (meaning of unmatched affluence), can be transformed into ‘āśamarddhiḥ’ because of the operation of a euphonic conjunction with a word ending in ‘a’ preceding it, or ‘āśamarddhi’ or ‘āsamarddhi’ in combination with words occurring after it or ‘asamardddhiḥ’ or ‘asamarddddhiḥ’ by internal transformation. Clearly, simply searching for the word asamarddhi would not yield the occurrences of the same word as ‘asamarddhi’, ‘āsamarddhi’, or other alternative forms. As such, a normal text-search using a Unicode text editor would not suffice. Other search engines currently used for Sanskrit [13] too do not provide for such comprehensive searching.

In order to achieve such an exhaustive search, all possible forms of the word resulting from the euphonic conjunctions that would become operative in its case must be generated and searched for in the given text.

The authors have already presented a new schema for fast sandhi processing in earlier work [4]. The present work extends the application of that schema to other sandhi rules including
consonant-based and visarga-based sandhis as well as important rules with respect to exceptional cases. It further presents a complete computational algorithm to process all sandhis, and an algorithm to apply this sandhi-processing procedure to generate all word forms to enable comprehensive searching.

A. Language Representation

The Unicode hexadecimal range 0900 - 097F is used to represent Sanskrit characters in Devanāgarī script. The characters used to represent Sanskrit letters in English script are found in the Basic Latin (0000-007F), Latin-1 Supplement (0080-00FF), Latin Extended-A (0100-017F) and Latin Extended Additional (1E00 – 1EFF) Unicode ranges.

The Latin character set has been employed in this work to represent Sanskrit letters as E-text. As such, the schema and algorithms presented do not use Devanāgarī script. To use the algorithms for text that is in Devanāgarī script, the text needs to first be converted to Latin text.

B. Terminology

The terminology employed in this work for certain groups of letters of the Sanskrit alphabet is given in Table 1.

| Term     | Description / Notation |
|----------|------------------------|
| Vowel    | a, ā, i, u, ā, r, ṛ, e, ae, o, au |
| Semi-vowel | y, v, r, l |
| Consonant | k, kh, g, gh, n, c, ch, j, jh, n, t, th, d, dh, n, t, th, d, dh, n, p, ph, b, bh, m, s, s, s, h |
| Guttural | k, kh, g, gh, n |
| Palatal  | c, ch, j, jh, ṛ, n |
| Cerebral | t, th, d, dh, n |
| Dental   | t, th, d, dh, n |
| Labial   | p, ph, b, bh, m |
| Nasal    | n, ŋ, n, m |
| Aspirate | h |
| Sibilant | s, ś, s |
| Column1  | k, c, t, t, p |
| Column2  | kh, ch, th, th, ph |
| Column3  | g, j, d, b |
| Column4  | gh, jh, dh, d, bh |
| Visarga  | ḷ |
| Anusvāra | m |
| Hard consonant | Column1, Column2, Sibilants |
| Soft consonant | Column3, Column4, Nasals, Aspirate |
| Hard guttural | k, kh |
| Hard labial | p, ph |
| Mutes    | Column1, Column2, Column3, Column4, Nasals |
| Jihvāmūlīya | (pronounced as the end of ‘kah’) |
| Upadhmānīya | (pronounced as the end of “paf”)

III. THE BASIS OF THE WORK

The renowned ancient Sanskrit linguist, Pāṇini, codified the extant grammar of Sanskrit into terse aphorisms (‘sūtras’) and organized these aphorisms into eight chapters. This work is the authoritative Aṣṭādhyāyī (literally meaning ‘work in eight chapters’) and is universally acknowledged as the most comprehensive codification of the grammar of any language. The grammatical rules that make up Pāṇini’s Aṣṭādhyāyī are derivational and known for their mathematical precision in spite of dealing with the nuances of the language at various levels including morphology, syntax, semantics, phonology and pragmatics. Owing to the cryptic nature of the Aṣṭādhyāyī, one or more of the commentaries on it are required to get a clear understanding of its contents.

The current work deals with Pāṇini’s sandhi-related aphorisms with the help of the recognized commentaries, Siddhānta-kaumudi [1] and Kāśikā [2]. Both these commentaries are accepted by Sanskrit scholars as authoritative works on Pāṇini grammar.

Pāṇini’s statements of grammatical rules are expressed on the basis of the Māheśvara-sūtras, or the ‘aphorisms of Mahēśvara’. These aphorisms provide a list of all the letters of the Sanskrit alphabet ordered in a specific sequence. The Māheśvara aphorisms are given below:

1. a-i-u-ṇ
2. r-[l-k
3. e-o-ṇ
4. ai-au-c
5. ha-ya-va-ṛ
6. la-ṇ
7. ṇa-ma-ṇa-na-m
8. jha-bha-ṇ
9. gha-dha-ḥa-s
10. j-a-ba-qa-da-da-ṣ
11. kha-pha-ḥa-ḥa-ca-ta-ta-v
12. ka-pa-y
13. sa-ṣa-sa-r
14. ha-l

The last letter in each of these aphorisms is only a placeholder. The first four aphorisms list only the short forms of all the vowels, while the rest list the semi-vowels and consonants; the latter list has the vowel ‘a’ appended to each letter only to enable pronunciation of the aphorism.

A. The Approach

The present work is based on earlier work by the authors, which directly codifies Pāṇini’s rules in a novel way using binary representations [4]. The unique data representation devised by the authors has been further refined in this work and consonant-based, visarga-based sandhi rules, as well as some special sandhi rules have been included in this work.

Rule representation has been simplified to minimal binary set-unset operations. Further, the sūtra ordering has been done after acquiring a thorough understanding of the operation of Pāṇini’s sandhi-related aphorisms. As such, this work presents a significant extension, refinement and closure of the earlier work of the authors. Moreover, it provides a clear understanding of the rules governing sandhi as laid down by Pāṇini, in a comprehensive and simplified way, hitherto not encountered in the literature.

B. The Binary Schema

The following is an extract from already published work by the authors [4] and is included here for completeness of the presentation.

http://sites.google.com/site/ijcsis/
A point of sandhi is denoted by
\[ x + y \]
where \( x \) and \( y \) denote the sandhi letters and the symbol ‘+’ denotes adjacency. The variable \( X \) denotes the sequence of letters culminating in \( x \); the variable \( Y \) denotes the sequence of letters starting with \( y \). The notations \( X \) and \( Y \) are used to depict special conditions that pertain to an entire word or sequence of letters involved in the sandhi rule. The letter immediately preceding \( x \) and the letter immediately succeeding \( y \) are denoted respectively as \( u \) and \( w \) respectively.

The refined schematic developed in this work to represent letters of the Sanskrit alphabet is given in Table 2.

Table 2: Binary representation scheme

| #  | Letters                                                                 |
|----|------------------------------------------------------------------------|
| 0  | a,ā,i,ī,u,ū,ṛ,ṝ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,��,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,ṝ,ḷ,ḷ, ṛ,��,ṝ,ḷ,ḷ, ṛ,��,ṝ,ḷ,ḷ, ṛ,��,ṝ,ḷ,ḷ, ṛ,뱅,ṝ,ḷ,ḷ, ṛ,뱅,ṝ,ḷ,ḷ, ṛ,뱅,ṝ,ḷ,ḷ, ṛ,뱅,��,ṝ,ḷ,ḷ, ṛ,뱅,��,��,��,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,뱅,
Algorithm SandhiRulesParser
{
    while Set 1 rules have not been fully visited
    {
        Try the next rule in Set 1;
        if the rule applies
        {
            Apply the rule and store the output;
            if the rule is optional
            {
                Add the current word pair to the word_list;
                Continue checking from the next rule for all
                word pairs stored in the word_list;
            }
            else
            {
                Process internal sandhi rules of Set 2;
            }
        }
    }
    while Set 2 rules have not been fully visited
    {
        Try the next rule in Set 2;
        if the rule applies
        {
            Apply the rule and transform the given words;
            if the rule is optional
            {
                Add the current word pair to the word_list;
                Continue checking from the next rule for all
                word pairs stored in the word_list;
            }
        }
    }
}

There are a few exceptions that would apply to the general processing order prescribed by the above algorithm. For example, in Set 1 the output produced by an optionally applying rule does, in certain cases, have to pass through a rule appearing below and undergo further transformation as a result, as it happens in Set 2. Also, it is found that a few rare rules of Set 2 have to be processed before Set 1. Further, in Set 2, all rules that form exceptions to a particular rule are stated after it by Pāṇini, but clearly, have to be processed before the rule by the algorithm.

V. THE SANDHI PROCESSOR

The key to symbols used in rule coding and algorithm specification is as follows:

- // means single-line explanatory comment
- { } are block or set indicators
- ∨ denotes or
- − denotes not
- ⊕ denotes xor
- | denotes word concatenation

The algorithm SandhiProcessor processes the rules pertaining to all the major sandhis in Sanskrit grammar, in accordance with the processing scheme presented in Algorithm SandhiRulesParser. Set 1 and Set2 sandhis have been incorporated here one below the other and the rules have been codified as per the schema presented above. The vowel sandhis presented in [4] have been modified in accordance to the reduced schema and included here for completeness.

When \( x_i, y_i \), etc., for \( i = 1, 2 \), are assigned new values by setting bits, it is assumed that their initial values are first unset. Also, if either part of a variable is not set, it is assumed to remain unchanged. Further, it is also assumed that a category change caused by a sandhi will cause an automatic change in the first four bits of the letter representation and that all bit representations for the changed letter will be generated thereafter. Hence, these aspects are not explicitly stated for each rule in this algorithm.

The speed of processing is increased by going into a rule only if overall conditions are satisfied. For instance, if the rule is a vowel sandhi rule, where both \( x \) and \( y \) are required to be vowels, then the check if \( x_i (0) \) and \( y_i (0) \) are 1 is first made. If this bit-check is not true for the input words, then a whole set of vowel sandhis is omitted from the parse, thus increasing the efficiency of the algorithm. These overall checks have not been shown in the algorithm presented below, in order to make the presentation more simple.

Algorithm SandhiProcessor \((X,Y)\)
{
    //1. svaujasamauçtaṣṭāhyāṁbhishiebhāṁbhayasāsasi
    //bhāṁbhayasāsasāmīyovsupsup || 4.1.2 ||
    //If there is a visarga (h) at the end of \( X \), then the visarga is
    //changed to ‘s’.
    if \( x_i (44) \)
    {
        \( x_i (24) = 1; \)
    }

    //2. sasajusru rh || 8.2.66 ||
    // Common name: visarga-rutva sandhi
    //If last letter of \( X \) is s, then s is replaced by ‘#’ which
    //stands for the particle ‘ru’, interpreted as ‘r’.
    // This rule is incorporated here though it belongs to the
    //sapādasaptādhyāyī.
    if \( x_i (24) \)
    {
        \( x_i (46) = 1; \)
    }

    //3. avaṅ sphaṭāyanasya || 6.1.123 || (vowel sandhi)
    // Common name: avaṅādeśa sandhi
    //If the word go is followed by a vowel, then the o of go
    //is optionally replaced by ava
    if \( (x_i (15) \land x_2 (0)) \land (u_1 (31) \land u_2 (3)) \)
    {
        Add \( X \) to word_list;
        \( x_i (19) = 1; \)
    }
}
4. ěsī ca || 6.1.114 ||
//If # (ru) or r at the end of X is preceded by the vowel a
//and followed by aspirate, semi-vowel, nasal, Column3 or
//Column4, then last letter of X is replaced with the vowel
///’u’ and shifted to Y to become the first letter of Y
if (x₁(46) ∨ x₁(21)) ∧ u₁(4) ∧ (y₁(1) ∨ y₁(26) ∨ y₁(27) ∨ y₁(38) ∨ y₁(39))
{ x₁(6) = 1;
 x₂ = u₂;
 Shift x from the end of X to the beginning of Y;
}

5. ato roraplaṭdaphule || 6.1.113 ||
//Common name: visarga-rutva sandhi
//If # (ru) or r at the end of X is followed and preceded by
//a, then the # or r is replaced with the vowel ’u’ and shifted
//to Y to become the first letter of Y
if (x₁(46) ∨ x₁(21)) ∧ u₁(4) ∧ y₁(4)
{ x₁(6) = 1;
 x₂ = u₂;
 Shift x from the end of X to the beginning of Y;
}

6. ēnāh padāntādati || 6.1.109 || (vowel sandhi)
//Common name: pūrvarūpa sandhi
//If e or o at the end of a word is followed by a, then e or o
//remains, and the avagraha () replaces a.
if x₁(13) ∧ y₁(4)
{ y₁(45) = 1;
}

7. akāh savarā dirgha || 6.1.101 || (vowel sandhi)
//Common name: savarnādirgha sandhi
//If one of a, i, u, r or s or their long equivalents ā, ī, ū and ū is followed by the short or long form of the same letter,
//then the corresponding long letter replaces both.
if (x₁(8) ∨ x₁(9)) ∧ (y₁(8) ∨ y₁(9)) ∧ ¬(x₂ ∨ y₂)
{ delete y;
 x₁(9) = 1;
 return X|Y;
}

8. omānaśca || 6.1.95 || (vowel sandhi)
//Common name: pararūpa sandhi
//If a or ā is followed by o of the word om or oṁ, then o
//replaces both.
if (x₁(4) ∨ x₁(5)) ∧ Y ∈ {om, oṁ}
{ delete x;
}

9. etyedhātyūthsu || 6.1.89 || (vowel sandhi)
//Common name: vrddhi sandhi
//For this rule, in all cases the resultant letter replaces x

//and y.
//i) If a or ā is followed by eti or edhati, then vrddhi letter
//ii) If the preposition pra is followed by esa or esy,
//then vrddhi letter ai replaces both
//iii) If a or ā is followed by āūh, then vrddhi letter au
//replaces both
//iv) If preposition pra is followed by ādāh, then vrddhi
//letter au replaces both
//v) If word sva is followed by šr, then vrddhi letter ai
//replaces both
if (x₁(4) ∨ x₁(5)) // is ‘a’ or ‘ā’
{ if (y₁(13) ∧ y₂(1)) // is ‘e’

//i) If Y starts with {et, edhat} // (i)
{ delete x;
 y₁(14) = 1;
 }

//ii) If Y starts with {es, esy} // (ii)
{ delete x;
 y₁(14) = 1;
 }

//iii) If Y starts with {ādah} // (iii)
{ delete x;
 y₁(14) = 1;
 }

//iv) If Y starts with {sva} // (iv)
{ delete x;
 y₁(14) = 1;
 }

//v) If Y starts with {pras, ava, apa, upa, parā} // (v)
{ delete x;
 y₁(14) = 1;
 }

//10. eṇi pararūpa || 6.1.94 || (vowel sandhi)
//Common name: pararūpa sandhi
//If a or ā at the end of a preposition is followed by e or
//o, then the e or o replaces both.
//Note: The prepositions that qualify are: pra, ava, apa,
//upa, parā.
if X ∈ {pra, ava, apa, upa, parā} ∧ y₁(13)
{ delete x;
}

//11. upasargādṛti dhātau || 6.1.91 || (vowel sandhi)
//Common name: vrddhi sandhi
//If a or ā at the end of a preposition is followed by r,
//i or l, then vṛddhi letter ār, ār or āl respectively //replaces both. Note: The prepositions that qualify are: //pra, parā, apa, ava, upa //ii) If the word vatsara, kambala, vasana, daśa, ṛṇa is //followed by the word ṛṇa, then vṛddhi letter ār //replaces both. //Note: This rule clashes with 6.1.87 (guna sandhi), and //takes precedence.

if \( X \in \{ \text{pra, ava, apa, upa, parā} \} \land y_1(12) \)

\[
\begin{align*}
\text{delete} & \; x; \\
y_1(17) & = 1;
\end{align*}
\]

if \( X \in \{ \text{vatsara, kambala, vasana, daśa, ṛṇa} \} \land \\
(y_1(12) \land y_2(0)) \land Y = \text{‘ṛṇa’} \)

\[
\begin{align*}
\text{delete} & \; x; \\
y_1(17) & = 1;
\end{align*}
\]

//12. vṛddhireci || 6.1.88 || (vowel sandhi) //Common name: vṛddhi sandhi //If a or ā is followed by e, o, ai or au, then the //corresponding vṛddhi letter ai or au replaces both. //Then \( x_1(4) \lor x_1(5) \lor (y_1(13) \lor x_1(14)) \)

\[
\begin{align*}
\text{delete} & \; x; \\
y_1(14) & = 1;
\end{align*}
\]

//13. ādgunah || 6.1.87 || (vowel sandhi) //Common name: guṇa sandhi //If a or ā is followed by i, ī, u, ū, ā or āl, then the //corresponding guṇa letter e, o, ar or al replaces both. //Then \( x_1(4) \lor x_1(5) \lor (y_1(10) \lor y_1(11)) \)

\[
\begin{align*}
\text{delete} & \; x; \\
y_1(16) & = 1;
\end{align*}
\]

//14. ecoyavāyāvah || 6.1.78 || (vowel sandhi) //Common name: ayāvāyāvādesa sandhi //If e, o, ai or au is followed by a vowel, then ay, av, āy, //āv replace the first respectively. //Then \( x_1(15) \land y_1(0) \)

\[
\begin{align*}
x_1(18) & = 1;
\end{align*}
\]

//15. iko yaņaci || 6.1.77 || (vowel sandhi) //Common name: yanādesa sandhi //If i, ī, u, ū, r, ā or l is followed by a vowel, then the //corresponding semi-vowel \((y, v, r, l)\) replaces the first. //Then \( x_1(10) \lor x_1(11) \lor y_1(0) \)

\[
\begin{align*}
x_1(20) & = 1;
\end{align*}
\]

//16. che ca || 6.1.73 || //Common name: tugāgama sandhi //If a short vowel is followed by the consonant ch, then //l is added. //Then \( x_1(8) \land (y_1(40) \land y_2(0)) \)

\[
\begin{align*}
z_1(34) & = 1; \\
z_2 & = y_2; \\
\text{Add} & \; z \; \text{to the end of} \; X;
\end{align*}
\]

//17. āṇmānoṣca || 6.1.74 || //Common name: tugāgama sandhi //If the particle ā or word mā is followed by ch, //then t is added. //Then \( X \in \{ ā, mā \} \land (y_1(40) \land y_2(0)) \)

\[
\begin{align*}
z_1(34) & = 1; \\
z_2 & = y_2; \\
\text{Add} & \; z \; \text{to the end of} \; X;
\end{align*}
\]

//18. dirghāt || 6.1.75 || //padāntādva || 6.1.76 || //Common name: tugāgama sandhi //If a long vowel is followed by ch, then t is added. //Then \( x_1(9) \land (y_1(40) \land y_2(0)) \)

\[
\begin{align*}
z_1(34) & = 1; \\
z_2 & = y_2; \\
\text{Add} & \; z \; \text{to the end of} \; X;
\end{align*}
\]

//19. sanīyogāntasya lokah || 8.2.23 || //If the final consonant of X is preceded by a //consonant, then the last consonant is dropped. //Then \( x_1(2) \land u_1(2) \)

\[
\begin{align*}
\text{delete} & \; x;
\end{align*}
\]

//20. jhalām jaśa’nte || 8.2.39 || //Common name: jaśva sandhi //If x is Column1, Column2, Column3, Column4, sibilant //or aspirate, then x is replaced by the corresponding //Column3. //Note: The rule sasajusru ruh || 8.2.66 || debars the //application of this rule for words ending in sibilants, and //has been incorporated earlier in Set 1 rules itself. //Then \( x_1(36) \lor x_1(37) \lor x_1(38) \lor x_1(39) \)

\[
\begin{align*}
x_1(38) & = 1;
\end{align*}
\]

//21. pumāk khayyampare || 8.3.6 || //atramunāsikāh pūrvasya tu vā || 8.3.2 || //If X is the word ‘pum’ or ‘pum’ and is followed by
//Column1 or Column2, which is in turn followed by a
//vowel, semi-vowel or a nasal, then x is replaced by #
//(ru) and the preceding vowel is made nasal using the
//anusvāra.
if $X \in \{pum, puṁ\} \land (y_1(36) \lor y_1(37)) \land (w_1(0) \lor w_1(1) \lor w_1(27))$
  {
    $x_1(43) = 1$;
    $z_1(46) = 1$;
    $x_2(0) = z_2(0) = 1$;
  }

//22. naśchavyapaśān || 8.3.7 ||
// atrānāsikah pūrvasya tu vā || 8.3.2 ||
//Common name: satva sandhi
//If the final n of a word, except for the word praśān,
//followed by ch, th, c, j or t which is in turn
//followed by a vowel, semi-vowel or nasal, then
//n is replaced with # (ru) and the preceding vowel is
//made nasal using the anusvāra.
if $\neg(X = \text{`praśān'}) \land (x_1(27) \land y_1(40)) \land (w_1(0) \lor w_1(1) \lor w_1(27))$
  {
    $u = x$;
    $x_1(43) = 1$;
    $z_1(46) = 1$;
    $x_2(0) = z_2(0) = 1$;
  }

//23. nīṃpe || 8.3.10 ||
// atrānāsikah pūrvasya tu vā || 8.3.2 ||
//If X = ‘nīṃ’ and y is the letter ‘p’ then x is
//optionally replaced with # (ru) and the preceding
//vowel is made nasal using the anusvāra.
if $X = \text{‘nīṃ’} \land (y_1(35) \land y_2(0))$
  {
    Add X|Y to word_list;
    $u = x$;
    $x_1(43) = 1$;
    $z_1(46) = 1$;
    $x_2(0) = z_2(0) = 1$;
  }

//24. ro ri || 8.3.14 ||
// dhralope pūrvasya dīrgho’ṇah || 6.3.111||
//If r or # (ru) is followed by r and is preceded by a, i or u,
//then one r is dropped and the short vowel is made long.
if $(x_1(46) \lor x_1(21)) \land y_1(21)$
  {
    delete x;
    if $u_1(6)$
      {
        $u_1(7) = 1$;
      }
    if $u_1(4)$
      {
        $u_1(5) = 1$;
      }
  }

//25. kharavasānyorvisarjanīyah || 8.3.15 ||
//If x is # (ru) or r and is followed by a hard consonant,
//then x is replaced with visarga.
if $(x_1(46) \lor x_1(21)) \land (y_1(36) \lor y_1(37) \lor y_1(23) \lor y_1(24) \lor y_1(25))$
  {
    $x_1(44) = 1$;
  }

//26. bhohagoahoapūrvasya yo ‘śi || 8.3.17 ||
//If x is # (ru) or r and is preceded by bho, bhago, agho,
//a or ā, and is followed by a vowel, semi-vowel or soft
//consonant, then x is replaced by the consonant ‘y’.
if $(x_1(46) \lor x_1(21)) \land (X_{–}(x)) \land \{bho, bhago, agho, a, ā\} \land (y_1(0) \lor y_1(1) \lor y_1(38) \lor y_1(39) \lor y_1(26) \lor y_1(27))$
  {
    $x_1(20) = 1$;
    $x_2(1) = 1$;
  }

//27. lopah sākalyasya || 8.3.19 ||
//If the consonant ‘y’ or ‘v’ is preceded by a or ā and is
//followed by a vowel, semi-vowel or soft consonant,
//then the ‘y’ or ‘v’ is dropped.
if $(x_1(20) \land x_2(0) \lor x_2(1))) \land (u_1(4) \lor u_1(5)) \land
(y_1(0) \lor y_1(1) \lor y_1(38) \lor y_1(39) \lor y_1(26) \lor y_1(27))$
  {
    delete x;
  }

//28. oto gārgyasya || 8.3.20 ||
//If the consonant ‘y’ is preceded by the vowel ‘o’ and
//followed by a vowel, semi-vowel or soft consonant,
//then the ‘y’ is dropped.
if $(x_1(20) \land x_2(1)) \land (u_1(3) \lor u_2(0)) \land
(y_1(0) \lor y_1(1) \lor y_1(38) \lor y_1(39) \lor y_1(26) \lor y_1(27))$
  {
    delete x;
  }

//29. uñi ca pade || 8.3.21 ||
//If the consonant ‘y’ or ‘v’ is preceded by a or ā and is
//followed by the word ‘u’, then the ‘y’ or ‘v’ is dropped.
if $(x_1(20) \land (x_2(0) \lor x_2(1))) \land (u_1(3) \lor u_1(5)) \land
Y = \text{‘u’}$
  {
    delete x;
  }

//30. hali sarvesāṁ || 8.3.22 ||
//If the consonant ‘y’ is followed by a semi-vowel or
//consonant, then the ‘y’ is dropped.
```cpp
//31. he mapare vā || 8.3.26 ||
//If \( m \) is followed by \( h \) at the end of a word which is in
//turn followed by \( m \), then the first \( m \) is optionally changed
//to anusvāra.
//ii) If \( m \) is followed by \( h \) which is in turn followed by
//consonants 'y', 'l', or 'v', then the \( m \) is optionally replaced
//by the nasal forms of 'y', 'l', or 'v' respectively.
if \( x_1(30) \land y_1(26) \)
{
  if \( w_1(30) \)
  {
    Add \( x \| Y \) to word_list;
    \( x_1(43) = 1; \)
  }
  else if \( w_1(20) \)
  {
    \( x_1(22) = 1; \)
    \( x_2 = w_2; \)
  }
}

//32. napare nah || 8.3.27 ||
//If \( m \) is followed by \( h \) at the end of a word which is in turn
//followed by \( n \), then the \( m \) is optionally replaced by \( n \).
if \( x_1(30) \land y_1(26) \land (w_1(27) \land w_2(2)) \)
{
  Add \( x \| Y \) to word_list;
  \( x = w; \)
}

//33. naścāpadāntasya jhali || 8.3.24 ||
//If \( n \) is followed by Column1, Column2, Column3,
//Column4, sibilant or aspirate not at the end of a word,
//then the \( n \) is replaced by anusvāra.
for (each letter \( x \) in a word and its succeeding letter \( y \))
{
  if \( (x_1(29)) \land (y_1(36) \lor y_1(37) \lor y_1(38) \lor y_1(39) \lor y_1(23) \lor y_1(24) \lor y_1(25) \lor y_1(26)) \)
  {
    \( x_1(43) = 1; \)
  }
}

//34. mo rājī samah kvau || 8.3.25 ||
//If \( m \) of the word 'sam' or 'sām' is followed by a word
//starting with 'rāj' or 'rāṭ' or 'rāṇ', then the \( m \) remains
//unchanged.
if \( X \in \{ \text{sam, sām} \} \land Y \in \{ \text{rāj, rāṭ, rāṇ} \} \)
{
  //Skip Rule 35
  Continue processing from Rule 36;
}

//35. mo'nuśvārah || 8.3.23 ||
//Common Name: anusvāra sandhi
//If \( m \) at the end of a word is followed by any consonant,
//then \( m \) is replaced by anusvāra.
if \( x_1(30) \land y_1(2) \)
{
  \( x_1(43) = 1; \)
}

//36. niṇoh kik tuḥ śāri || 8.3.28 ||
//If \( n \) or \( n \) is followed by a sibilant, then \( k \) or \( t \) is optionally
//added respectively.
if \( x_1(28) \land (y_1(23) \lor y_1(24) \lor y_1(25)) \)
{
  Add \( X \| Y \) to word_list;
  \( z_1(36) = 1; \)
  \( z_2 = x_2; \)
  Add \( z \) to the end of \( X; \)
}

//37. daḥ si dhū || 8.3.29 ||
//Common name: dhūdaṅgama sandhi
//If \( d \) is followed by \( s \), then \( dh \) is added optionally
if \( x_1(38) \land x_2(1) \land y_1(24) \)
{
  Add \( X \| Y \) to word_list;
  \( z_1(34) = 1; \)
  \( z_2 = x_2; \)
  Add \( z \) to the end of \( X; \)
}

//38. naṣca || 8.3.30 ||
//Common name: dhūdaṅgama sandhi
//If \( n \) is followed by \( s \), then \( dh \) is added optionally.
if \( x_1(27) \land x_2(2) \land y_1(24) \)
{
  Add \( X \| Y \) to word_list;
  \( z_1(39) = 1; \)
  \( z_2 = x_2; \)
  Add \( z \) to the end of \( X; \)
}

//39. śi tuḥ || 8.3.31 ||
//Common name: tugāgama sandhi
//If \( n \) is followed by \( s \), then \( t \) is optionally added.
if \( x_1(27) \land x_2(2) \land y_1(25) \)
{
  Add \( X \| Y \) to word_list;
  \( z_1(36) = 1; \)
  \( z_2 = x_2; \)
  Add \( z \) to the end of \( X; \)
}

//40. naḥhrasvadaci ṇaṇumānyiṇaḥ || 8.3.32 ||
//Common name: naḥhrasvāda sandhi
//If \( n \), \( n \) or \( n \) is preceded by a short vowel and succeeded by
//a vowel, then the \( n \), \( n \) or \( n \) gets duplicated.
```

\[
\text{if } (x_1(27) \land (x_2(0) \lor x_2(1) \lor x_2(2))) \land u_1(8) \land y_1(0) \\
\{ \\
\text{Add another x to the end of X;}
\}
\]

\[
/41. \text{ śarpāre visarjaniyōḥ} \| 8.3.35 \|
\text{If visarga is followed by a hard consonant which is in turn} \\
\text{followed by a sibilant, then the visarga is retained.} \\
\text{if } x_1(44) \land (y_1(36) \lor y_1(37) \lor y_1(23) \lor y_1(24) \lor y_1(25)) \land (w_1(23) \lor w_1(24) \lor w_1(25)) \\
\{ \\
\text{Store result with no change}
\}
\]

\[
/42. \text{ vā śari} \| 8.3.36 \|
\text{If visarga is followed by a sibilant, then the visarga is} \\
\text{optionally retained.} \\
\text{if } x_1(44) \land y_1(41) \\
\{ \\
\text{Add X|Y to word_list;}
\text{delete x;}
\}
\]

\[
/43. \text{kupvoh ḅā ṣpau ca} \| 8.3.37 \|
\text{If visarga is followed by a hard guttural or hard labial,} \\
\text{then it is replaced optionally by } \land \text{ (pronounced as at the end of ‘kah’)} \\
\text{or } \lor \text{ (pronounced as at the end of ‘paf’)} \\
\text{if } x_1(44) \land y_1(41) \\
\{ \\
\text{Add X|Y to word_list;}
\text{x_1(42) = 1;}
\text{x_2 = y_2;}
\text{Flag_kupvoh_sutra_fired = true} //flag is set
\}
\]

\[
/44. \text{ so’padādau} \| 8.3.38 \|
\text{If visarga is followed by pāṣa, kalpa, ka or kāmya, then} \\
\text{the visarga is replaced by s.} \\
\text{if } x_1(44) \land Y \text{ begins with } \{pāṣa, kalpa, ka, kāmya\} \\
\{ \\
\text{x_1(24) = 1;}
\}
\]

\[
/45. \text{ inah śah} \| 8.3.39 \|
\text{If visarga is preceded by ‘i’ or ‘u’ and followed by pāṣa,} \\
\text{kalpa, ka or kāmya, then the visarga is replaced by s.} \\
\text{if } x_1(44) \land u_1(6) \land Y \text{ begins with } \{pāṣa, kalpa, ka, kāmya\} \\
\{ \\
\text{x_1(23) = 1;}
\}
\]

\[
/46. \text{ namaspurasorgatyoh} \| 8.3.40 \|
\text{If ‘namah’ or ‘purah’ is followed by a hard guttural or} \\
\text{hard labial, then the visarga is replaced by s optionally.} \\
\text{if } X \in \{\text{namah, purah}\} \land y_1(41) \\
\{ \\
\text{Add X|Y to word_list;}
\text{x_1(24) = 1;}
\}
\]

\[
/47. \text{idadupadhaya cā’pratyayasya} \| 8.3.41 \|
\text{If visarga is preceded by ‘i’ or ‘u’ and is at the end of any} \\
\text{or nih, duh, bahih, āvih, catuh, práduḥ} \text{ and is followed by} \\
\text{a hard guttural or hard labial, then the visarga is replaced} \\
\text{by s.} \\
\text{if } X \in \{\text{nih, duh, bahih, āvih, catuh, práduḥ}\} \land y_1(41) \\
\{ \\
\text{x_1(23) = 1;}
\}
\]

\[
/48. \text{tiraso’nyatarasyāṃ} \| 8.3.42 \|
\text{If the word ‘tirah’ is followed by a hard guttural or hard} \\
\text{labial, then the visarga is optionally replaced by s.} \\
\text{if } X = \text{‘tirah’} \land y_1(41) \\
\{ \\
\text{Add X|Y to the word_list;}
\text{x_1(24) = 1;}
\}
\]

\[
/49. \text{dvistriścaturiti kṛtvorthe} \| 8.3.43 \|
\text{If the words dvī, trih or catuḥ are followed by a hard} \\
\text{guttural or hard labial, then the visarga is optionally} \\
\text{replaced by s.} \\
\text{if } X \in \{\text{dvī, trih, catuḥ}\} \land y_1(41) \\
\{ \\
\text{Add X|Y to the word_list;}
\text{x_1(23) = 1;}
\}
\]

\[
/50. \text{atah kṛkamikaṁśakumbhāpātratrukṣā} \\
\text{karnīṣvanavayasya} \| 8.3.46 \|
\text{If visarga is preceded by a and followed by a form of kr} \\
\text{or kam or by the words kāṁsa, kumbha, pātra, kuśā or} \\
\text{karnī, then the visarga is replaced by s.} \\
\text{if } x_1(44) \land u_1(4) \land Y \text{ begins with } \{\text{kr, kar, kur, kam, kāṁ, kaṁsa, kumbha, pātra, kuśā, karṇī}\} \\
\{ \\
\text{x_1(24) = 1;}
\}
\]

\[
/51. \text{adhah śirasī pade} \| 8.3.47 \|
\text{If the word adhah or śirah is followed by the word ‘pada’,} \\
\text{then the visarga is replaced by s optionally} \\
\text{if } X \in \{\text{adhah, śirah}\} \land Y \text{ begins with } \{\text{pad}\} \\
\{ \\
\text{Add X|Y to word_list;}
\text{x_1(24) = 1;}
\}
\]

\text{//Rules 41 to 51 form exceptions to the following rule, Rule} \\
\text{//52, and hence have been handled before it.}
//52. visarjanīyasya saḥ || 8.3.34
//If visarga is followed by a hard consonant, then s replaces
//the visarga.
if Flag_kupvoh_sutra_fired = false //8.3.37 is not fired
{
    if \( x_1(44) \land (y_1(36) \lor y_1(37) \lor y_1(23) \lor y_1(24)) \lor y_1(25) \)
    
    \( x_1(24) = 1; \)
}
}

//53. raśābhyaḥ no nahi samānapade || 8.4.1
//aṭkupvānumvyāvāye'pi || 8.4.2
//padāntasya || 8.4.37
//If n is preceded by r, ṛ, r or s within the same word, and a
//palatal, cerebral, dental, l, s or s does not lie between the
//two, and n is not the last letter of the word, then n is
//replaced by n.
for (each letter y in X where y is not the last letter)
{
    if \( y_1(27) \land y_2(2) \) //y is ‘n’
    
    if \( \exists \) a letter x in X preceding y where \( (x_1(12) \land
    \neg x_2(2)) \lor x_1(21) \lor x_1(23) \) //x is r, ṛ, r or s
    
    if \( \exists \) any letter q between x and y where \( q_1(32)
    \lor q_1(33) \lor q_1(34) \lor (q_1(20) \land q_2(3)) \lor
    q_1(24) \lor q_1(25)
    
    \( y_2(1) = 1; \)
    
    \}
}

//Repeat the above for Y

//54. stōḥ ścunāḥ ścuh || 8.4.40
//śār || 8.4.44
//Common name: ścavva sandhi
//If a palatal other than s is followed by a dental, or a dental
//is followed by a palatal, then the dental is replaced by the
//corresponding palatal.
if \( x_1(32) \land \neg x_2(5) \land (y_1(34))

\( y_1(32) = 1; \)

elseif \( x_1(34) \land y_1(32) \)

\( x_1(32) = 1; \)

//55. śtunāḥ stuh || 8.4.41
//na padāntāṭantarām || 8.4.42
//toḥ sti || 8.4.43
//Common name: śtutva sandhi
//If a dental is followed by a cerebral except s, or if the
//specific cerebral r is followed by nām, navatī or nagarī, or
//if a cerebral is followed by a dental, then the dental is
//replaced by the corresponding cerebral.
if \( (x_1(34) \land \neg x_2(5)) \land (y_1(33))

\{ 
    \( y_1(33) = 1; \)
    
    elseif \( x_1(33) \land x_2(0) \land Y \in \{nām, navat, nagar\}

\{ 
    \( y_1(33) = 1; \)
    
    elseif \( x_1(33) \land y_1(34)

\{ 
    \( y_1(33) = 1; \)
    
    //56. yaro'nunāsike'numāsiko vā || 8.4.45
//Common name: annāsikā sandhi
//If a consonant other than ‘h’ is followed by a nasal, then
//the consonant is optionally replaced by the corresponding
//nasal. The rule is obligatory if the second word is
//‘mayam’ or ‘mātram’.
if \( x_1(36) \lor x_1(37) \lor x_1(38) \lor x_1(39) \)

\{ 
    if \( y_1(27)

\{ 
    Add X|Y to the word_list;

    \( x_1(27) = 1; \)
    
    elseif \( Y \in \{maya, mātra\}

\{ 
    \( x_1(27) = 1; \)
    
    //57. aco rahāhbyām dve || 8.4.46
//If r or h is followed by any consonant other than h and
//preceded by a vowel, then the consonant is duplicated
//within a word.
for (each set of consecutive letters u,x,y in X)

\{ 
    if \( x_1(21) \lor x_1(26) \lor (y_1(2) \land \neg y_1(26)) \land u_1(0)

\{
    \( z = y; \)
    
    Add z after x;
    
    \}
    
    elseif \( x_1(21) \lor x_1(26) \lor (y_1(2) \land \neg y_1(26)) \land u_1(0)

\{
    \( z = y; \)
    
    Add z after x;
    
    \}
//58. anaci ca || 8.4.47 ||
// dirigha\nti\n\n\nIf any consonant other than h is preceded by a short vowel
// and followed by anything other than a vowel, then the
// consonant is doubled within a word.
for (each set of consecutive letters x,y,w in X)
{
  if \( x_1(8) \land (y_1(2) \land \neg y_1(26)) \land (w_1(1) \lor w_1(2) \lor w_1(3)) \)
    \[
    z = y;
    \]
    Add z after x;
}
for (each set of consecutive letters x,y,w in Y)
{
  if \( x_1(8) \land (y_1(2) \land \neg y_1(26)) \land (w_1(1) \lor w_1(2) \lor w_1(3)) \)
    \[
    z = y;
    \]
    Add z after x;
}

//59. jhal\mt j\nti jha\nti || 8.4.53 ||
// Common name: ja\nti\nti sandhi
// If a non-nasal mute, sibilant or aspirate is followed by
// Column3 or Column4, then the first letter is replaced by
// the corresponding Column1 letter.
for (each set of consecutive letters x, y in X)
{
  if \( y_1(38) \lor y_1(39) \)
    \[
    \]
    \[
    \]
  else if \( x_1(23) \lor x_1(24) \lor x_1(25) \)
    \[
    x_1(38) = 1;
    \]
    \[
    \]
    \[
    \]
    \[
    \]
    \[
    \]
//62. torli || 8.4.60 ||
//Common name: parasavarṇa sandhi
//i) If n is followed by l, then n is replaced by nasal l.
//ii) If a dental other than n and s is followed by l, then
//the dental is replaced by l.
if \((x_1(27) \land x_2(2)) \land (y_1(20) \land y_2(3))\)
{
    \(x_1(22) = 1;\)
    \(x_2 = y_2;\)
}
else if \((x_1(34) \land \lnot(x_2(4) \lor x_2(5))) \land (y_1(20) \land y_2(3))\)
{
    \(x = y;\)
}

//63. jhayoho’nyatarasyām || 8.4.62 ||
//Common name: pūrvasavarṇa sandhi
//If a non-nasal mute is followed by h, then h is optionally
//replaced by the Column4 letter corresponding to the non-
//nasal mute.
if \((x_1(36) \lor x_1(37) \lor x_1(38) \lor x_1(39)) \land y_1(26)\)
{
    Add \(X|Y\) to the word_list;
    \(y_1(39) = 1;\)
    \(y_2 = x_2;\)
}

//64. śascho’i || 8.4.63 ||
//Common name: chatva sandhi
//If a non-nasal mute is followed by ś which is in turn
//followed by a vowel, semi-vowel or nasal, then ś is
//optionally replaced by ch.
if \((x_1(36) \lor x_1(37) \lor x_1(38) \lor x_1(39)) \land (w_1(0) \lor w_1(1) \lor w_1(27))\)
{
    Add \(X|Y\) to the word_list;
    \(y_1(40) = 1;\)
}

//65. halo yamāṁ yami lopah || 8.4.64 ||
//If a semi-vowel or nasal is preceded by a consonant and
//followed by the same semi-vowel or nasal letter, then one
//of the duplicate letters is dropped.
if \(u_1(2)\)
{
    if \((x_1(20) \lor x_1(27)) \land y == x\)
    {
        delete x;
    }
}

//66. jharo jhari savarne || 8.4.65 ||
//If a non-nasal mute or sibilant is preceded by a consonant
//or semi-vowel and followed by a homogeneous mute or
//sibilant, then one of the duplicate letters is optionally
dropped, within a word
for (each set of consecutive letters \(u,x,y\) in \(X\))
{
    if \(u_1(1) \lor u_1(2)\)
    {
        if \(x_1(36) \lor x_1(37) \lor x_1(38) \lor x_1(39) \lor x_1(23) \lor x_1(24) \lor x_1(25)\) \land \(x_1 == y_1\)
        {
            Add \(X|Y\) to the word_list;
            delete x;
        }
    }
}
// Repeat the above for \(Y\)

C. The Search Engine

Algorithm SandhiProcessor may be used to generate alternative forms of a given search word. The following
algorithm is used to generate all possible alternative forms of a
given word, by providing possible word forms before and after
the word so that sandhi rules get triggered. All these word
forms are searched for in the E-text.

Algorithm GenerateAllWordForms (Z)
{
//\(Z\) is the search word.
//\{WordForms\} denotes the set of word forms generated by
//the algorithm and is initially the null set.
Add Z to \{WordForms\};
\(X = Z;\)
for (each \(y\) in \{vowels, semi-vowels, consonants\})
{
    Add SandhiProcessor(X, Y) to \{WordForms\};
}
for (each \(Y \in \{\text{WordForms}\}\))
{
    for (each \(X \in \{\text{vowels, semi-vowels, consonants, } h, m, #\})\)
    {
        Add SandhiProcessor(X, Y) to \{WordForms\};
    }
}

VI. CONCLUSION

The schema developed in this work presents a simple yet
unique and efficient method to process the sandhi aphorisms of
Pāṇini. The letter representation scheme is binary, and hence
all the checks are implemented as bit-level operations and
simple bit-set and bit-unset operations suffice to carry out the
sandhi transformation. The efficiency is further enhanced by
the division of a letter representation into two parts and the
consequent reduction of the transformation process to a shifting
of category. Further, this pattern of solving the sandhi
construction problem is unprecedented in the literature. Thus,
representation schema and the results of the sandhi-processing
algorithm represent an efficient computational model to process Sanskrit euphonic conjunctions. It must be mentioned here that some rules such as those with regard to prakṛtibhāva sandhi (non-transformational sandhi) have not been presented above. However, it is clear that their implementation is only an extension of the algorithm presented in this work that does not require any new schema.

The representational schema has been reduced further from [4] in this work, since many more sandhi rules have been incorporated here. The optimality of the schema is clear from the simplicity of the rule representation.

The final algorithm presented in this work, which uses this sandhi processor for word searches in E-texts is the first of its kind in the literature with regard to Sanskrit.

The use of the sandhi processor for searching ensures comprehensiveness of the search, while the efficiency of the sandhi processing method presented in this work ensures that search speeds are not compromised due to the increase in the number of words to be searched for. This was confirmed in the implementation of the algorithm.

The algorithms presented in this work have been tested for use with Sanskrit E-text in Devanāgarī script after a conversion engine converted Devanāgarī Unicode to Latin Unicode E-text.

REFERENCES
[1] Dīkṣita Bhaṭṭoji, Siddhānta-kaumudī, Translated by Śrīśa Candra Vasu, Volume 1, Motilal Banarsidas Publishers, Delhi, 1962.
[2] Vāmana & Jayāditya, Kāśikā, with the subcommentaries of Jinendrabuddhi, Haradatta Miśra and Dr. Jaya Shankar Lal Tripathi, Tara Book Agency, Varanasi, 1984.
[3] Rama N., Meenakshi Lakshmanan, A New Computational Schema for Euphonic Conjunctions in Sanskrit Processing, International Journal of Computer Science Issues, Vol. 5, 2009, pp 43-51 (ISSN print: 1694-0814, ISSN online: 1694-0784).
[4] Kasmir Raja S. V., Rajitha V., Meenakshi Lakshmanan, A Binary Schema and Computational Algorithms to Process Vowel-based Euphonic Conjunctions for Word Searches, International Journal of Applied Engineering Research, Volume 9, Number 20(2014) pp. 7127-7141 (ISSN print: 0973-4562, ISSN online: 1087-1090).

Websites
[5] Göttingen Register of Electronic Texts in Indian Languages (GRETIL), gretil.sub.uni-goettingen.de/gretil.htm
[6] Sanskrit Documents, sanskritdocuments.org
[7] Indology: Resources for Indological Scholarship, indology.info/etexts/archive/ etext
[8] TITUS, titus.uni-frankfurt.de/indexe.htm
[9] Muktabodha Indological Text Collection & Search Engine, muktalib5.org/digital_library_secure_entry.htm