

Sanskrit Compound Generation: With a Focus on the Order of Operations

**A dissertation submitted to the University of Hyderabad
for the award of the degree of**

Doctor of Philosophy

in

Sanskrit Studies

Pavankumar Satuluri

11HSPH04



Department of Sanskrit Studies

School of Humanities

University of Hyderabad

Hyderabad

July 2015

Sanskrit Compound Generation: With a Focus on the Order of Operations

**A dissertation submitted to the University of Hyderabad
for the award of the degree of**

Doctor of Philosophy

in

Sanskrit Studies

by

Pavankumar Satuluri

11HSPH04

under the guidance of

Dr. Amba P. Kulkarni



Department of Sanskrit Studies

School of Humanities

University of Hyderabad

Hyderabad

July 2015

Declaration

I, **Pavankumar Satuluri**, hereby declare that the work embodied in this dissertation entitled “ **Sanskrit Compound Generation: With a Focus on the Order of Operations**” is carried out by me under the supervision of Dr. Amba P. Kulkarni, Associate Professor, Department of Sanskrit Studies, University of Hyderabad, Hyderabad and has not been submitted for any degree in part or in full to this university or any other university. I hereby agree that my thesis can be deposited in Shodhganga/INFLIBNET.

A report on plagiarism statistics from the University Librarian is enclosed.

Pavankumar Satuluri
11HSPH04

Date:

Place: Hyderabad

Signature of the Supervisor



Department of Sanskrit Studies
University of Hyderabad, Hyderabad

Certificate

This is to certify that **Pavankumar Satuluri (11HSPH04)** has carried out the research-work embodied in the present dissertation entitled "**Sanskrit Compound Generation: With a Focus on the Order of Operations**" at University of Hyderabad. The dissertation represents his independent work and has not been submitted for any research degree of this university or any other university.

Dr. Amba P. Kulkarni

Supervisor

Associate Professor

Department of Sanskrit Studies

School of Humanities

Dr. J.S.R.A Prasad

Head

Associate Professor

Department of Sanskrit Studies

School of Humanities

Prof. Panchanan Mohanty

Dean

School of Humanities

University of Hyderabad

Acknowledgements

My sincere gratitude to my supervisor *Dr. Amba P. Kulkarni* for her guidance, patience and continuous support throughout my research work. I got an opportunity to do my research under her valuable guidance.

I express my utmost gratitude to *Prof. K. V. Ramakrishnamacharyulu* for their valuable suggestions and guidance throughout my entire research work.

I express my sincere gratitude to *Prof. K. Subrahmanyam* for their continuous support in profound discussions on various research problems.

I express my heartfelt thanks to *Prof. K. N. Murthy* for their continuous support, guidance and encouragements.

I express my sincere gratitude to *Dr. J. S. R. A Prasad* for their constant support and encouragement in my research.

I also convey my regards to my teacher *Dr. B. V. Venkataramana* who gave me valuable insights on my work.

I am indebted to *Dr. Anil Kumar, Sivasenani* for their help during my research.

I thank to all my friends *Arjun, Monali, Dr. Vinaya, Dr. Vani, Dr. Siva, Surendra, Krishna Mohan, Sanjeev, Madhusudan, Krishnamurthy, Gowri* and *Karunakar* for their helping and support.

I am most grateful to my *parents* for their continuous support and encouragement.

I cannot forget office staff of our department, who provided me all kind of infrastructural help. I thank one and all from Department of Sanskrit

Studies office.

I would like to thank one and all, who have directly or indirectly been instrumental in the completion of my research work and thesis.

Contents

Title Page	i
Declaration	ii
Certificate	iii
Acknowledgements	iv
Table of Contents	vi
List of Figures	viii
Thesis Related Papers Presented at International Conferences	ix
1 Motivation	1
1.1 Importance of Aṣṭādhyāyī	1
2 Compound Formation : Pāṇinian Procedure	12
3 Theoretical Issues in Compound Formation	19
3.1 When should subluk take place?	21
3.2 Meaning of the sūtra samāsāntāḥ (A5.4.68)	22
3.3 At what stage should the samāsānta suffix be added?	23

4	Analysis and Classification of Sūtras	30
5	Semantic and Pragmatic Information Needed for Compound Generation	56
5.1	List of jātis needed for Compound Generation	61
5.2	List of Upādhis needed for Compound Generation	66
5.3	Pragmatics	69
6	Implementation Details	73
6.1	Description of modules and their implementation	73
6.2	Scope of Generator	80
6.3	Some specific issues	81
6.4	Evaluation	82
7	Conclusion	83
	PART - II	86
	Bibliography	

List of Figures

3.1	Process of Compound Generation (1)	20
3.2	Process of Compound Generation (2)	29
5.1	Ontological classification	66
5.2	Other properties	68
6.1	Deciding the position of the component	78

Thesis Related Papers Presented at International Conferences

Pavankumar Satuluri and Amba Kulkarni, "Generation of Sanskrit Compounds". International Conference on Natural Language Processing December 19th-20th 2013. CDAC-Noida.

Pavankumar Satuluri, K. V. Ramakrishnamacharyulu and Amba Kulkarni, "Samāsaprakriyāyāṃ samāsāntānāṃ sthānaṃ, samāsanirdhāraṇe teṣāmupayogaśca" World Sanskrit Conference June 28th-July 2nd 2015. Bangkok, Thailand.

Chapter 1

Motivation

1.1 Importance of Aṣṭādhyāyī

Grammar plays an important role in the study of any language. It helps in the choice of the correct form of words. Pāṇini a great Indian grammarian compiled Aṣṭādhyāyī, a Sanskrit treatise on grammar, circa 4th century BC. Several Grammars for Sanskrit were available prior to Pāṇini. Pāṇini built upon this knowledge of his precursors and compiled them in the form of Aṣṭādhyāyī. The simplicity, thoroughness and the sheer scope of this work, according to Kornai (2008), would place Pāṇini among the greatest landmarks of all intellectual history even if we disregarded the key methodological advances he made. Aṣṭādhyāyī is an extant and almost full fledged grammar covering the then prevalent Sanskrit. To our knowledge this is the only grammar of any natural language with such a wide coverage. The way Pāṇini analysed Sanskrit has attracted several linguists towards it and we see the influence of Pāṇini's grammar since the development stages of the Western linguistics. We mention below a few quotes or the works that motivated us to choose our current topic of research.

Staal (1994)[page:2917] says that

“Pāṇini, 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”.

Leonard Bloomfield described Pāṇini’s Aṣṭādhyāyī as “One of the greatest monuments of human intelligence”.

Ferdinand de Saussure who is considered to be the father of modern structural linguistics was influenced by Pāṇini and Bhartṛhari. No wonder then that the western linguistics is influenced by Pāṇini’s grammar during its development stage.

The organisation of sūtras, use of ‘pratyāhārās’, use of ‘anubandhās’, use of ‘anuvṛtti’, use of his meta-language and ordering of rules have attracted the linguists as well as computer scientists.

Cardona (1969) observes that

“Patañjali explains, the ordering of sounds and the placing of anubandhas are both intended for the formation of pratyāhāras to be used in rules. This point is repeated by various commentators but with one addition; they specifically mention brevity (lāghava)”.

Kiparsky (2007)[page:1] states that

“Completeness and Economy are rightly considered the two main goals of Pāṇini’s grammar”.

The computer scientists were attracted on the one hand by this algorithmic style, and various methodologies used in Pāṇini, and on the other hand looked at the Aṣṭādhyāyī as a tool for information processing.

Gérard Huet in his inaugural speech at 1st International Sanskrit Computational Linguistics Symposium said that

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

Bharati and Kulkarni (2010) discuss a set of sūtras from Aṣṭādhyāyī which deal with various aspects of the information coding in sanskrit.

Petersen (2004), a German scholar was intrigued by the arrangement of sounds in the Śiva sūtras and provided a mathematical tool to decide if a set of partial ordered sets can be arranged in a linear order, and if yes, provide the linear order as well. She also proved mathematically that the repetition of the sound ‘h’ in the pratyāhāras is optimal, and further tried to provide Śivasūtras for the German sounds as well.

Joshi (2009) says that

“Pāṇini has intuitively used Calculus. Bhattojīdīkshita composed the book Siddhantakaumudī based on the Aṣṭādhyāyī which is a generative grammar also known as prakriyā. When we deal with the prakriyā part, it involves various stages. In every stage we face some transformations of stem or suffix and some extra augments will take place, and this type of procedure is equivalent to an algorithm”.

Several researchers have looked at the algorithmic aspect of Pāṇini and tried to develop simulators that will follow Aṣṭādhyāyī sūtra order.

Goyal et al. (2009) developed an Aṣṭādhyāyī simulator. In that they discuss the sūtras pūrvatrāsiddham (8.2.1) asiddhavadatrābhāt (6.4.22) ṣatvatukorasiddhaḥ (6.1.86) to know how these play a major role in ordering of the sūtras and how Pāṇini dealt with the conflict resolution techniques especially utsarga and apavāda in Aṣṭādhyāyī. This simulator triggers the rules and resolve the conflicts between the sūtras automatically.

Subbanna and Varakhedi (2010) have developed a computational model of Aṣṭādhyāyī based on the principle of asiddhatva. A mathematical concept called ‘filter’ is newly introduced to deal with all the usages of asiddhatva. Further, they have discussed about the computational aspects and conflict resolution techniques in Aṣṭādhyāyī.

Hyman (2009) observes that the Pāṇinian sūtras for external sandhi can be modeled using a finite state grammar. He has coded all the sandhi sūtras in XML format. This XML formalism has been chosen to express the rules in a machine readable form. He talks about a rule compiler, which translates the rules that are encoded in the XML formalism and converts these into some rewrite rules and those are compiled automatically into a Finite State Transducer (FST) by using some algorithms.

Some other efforts towards computerisation of Aṣṭādhyāyī are by Dhaval Patel and Swami Taralabalu.

The syntax of Pāṇini's sūtras was found to have striking similarity to the Backus Naur Form (BNF) that is used to describe the syntax of programming languages. BNF and chomsky's hierarchy of formal languages and grammar provided another dimension to look at Pāṇini's Aṣṭādhyāyī.

Bhate and Kak (1992) examined the context sensitive grammar for some rules in Aṣṭādhyāyī. Such as iko'yaṇaci (A6.1.74).

Staal (1965) observes that context free rules were clearly insufficient for Pāṇini's purpose. According to him, Pāṇini's Aṣṭādhyāyī is a mixture of context free and context sensitive rules.

Goal of research

Most of these works focussed on the sandhi formation and inflectional morphology. Sanskrit is also rich in derivational morphology and as such for any meaningful computational work, we need a generator that handles derivational morphology. Three important derivational processes that generate new nominal stems are kṛt, taddhita and samāsa operations. We decided to work on samāsa. We have a keen interest in understanding the concept of compounds at both prakriyā as well as ārthika level. For the usage, prakriyā part is very useful and ārthika level helps us to know the concepts like vṛtti and meaning compatibility (sāmarthya).

Compounds play a vital role in any language. Sanskrit is no exception to it. Compounds provide a compact way of expressing the thoughts. For the languages that are rich in compound formation, compound analysis and generation becomes an important and essential component of any Natural

Language Processing system. The reason is, in Sanskrit every sentence consists of one or more compounds. For example, look at the following śloka from Kālidāsa's Raghuvamśa

*vāgarthāvivā sampr̥ktau vāgarthapratipattaye
jagataḥ pitarau vande pārvatīparamēśvarau*

It contains seven words out of which four are compounds. We should know the meaning of the compounds in order to understand the meaning of the śloka. This is not an isolated instance of a śloka with several compounds. The analysis of a sample corpus of 400 thousand words show that every fifth word in the corpus is a compound.

In Indian grammatical tradition, there is an abundant discussion of compounds both at the level of *prakriyā* 'the compound formation involving morphology and phonology', and also at the level of *artha* 'meaning analysis'. In recent years, many scholars have worked on compounds on both of these aspects.

The *samarthāhnikā* of Joshi (1968) gives a detailed account of the discussion involved in the Indian tradition on the semantic compatibility of components and the compositionality of the meaning of a compound from its components. Further, he discusses the mutual compatibility of meaning of components and how the principle of gamakatva helps in understanding the meaning of compounds that violate the mutual meaning compatibility (asamartha padavidhi).

Pataskar (1996) has discussed the use of the Dvandva compounds in Pāṇini sūtras in relation to their case endings.

Bhandare (1995) has discussed the structural and semantic aspects of dvandva compound. He discusses the complete process of dvandva compound formation and observes that unlike other compounds Dvandva compounds enjoy some free word order among its components.

Mahavir (1986) talks about the process of generating a compound from its paraphrase (vigrahavākya). He has discussed various transformation rules that takes place on the vigrahavākya to get a compound form. Further, he discusses nityasamāśas and morphophonemic rules in great detail.

The semantic classification of compounds given by Pāṇini is not only restricted to Sanskrit language per say, but is universal. For example, the Cambridge grammar of the English Language uses this classification to describe compounds in English.

Compounds are always binary with an exception of conjunctive (Dvandva) and some exocentric (*Bahuvrīhi*) compounds. Individual components of the compounds can themselves be compounds, thus allowing a recursion in the formation of compounds. For understanding such compounds the underlying constituency structure is important.

Gillon (2009) proposes an extended phrase structure syntax to represent the underlying constituent structure of the compounds. He examined

the analysis of the non constituent compounds and exocentric compounds with the help of constituency grammar. He suggested a tagging scheme for compounds using context free rules, wherein he specifies the vibhakti and identifies the head of the compound and also provides the category of the compound.

Kumar (2012) has described the computational analysis of Sanskrit compounds in his doctoral dissertation. While this system uses Pāṇini's rules for identification of the type and analysis of exceptional compounds, the frequent compound type identification and analysis is done using the statistical properties of the corpus.

Goyal and Huet (2013) describes various morphological phenomena involved in the generation and analysis of *avyayībhāva* compounds.

We were fascinated by all this literature and decided to look at the process of compound formation in detail. The main reasons for choosing the domain of 'compounds' are:

- (a) There is a well laid down procedure for compound formation, with several examples worked out to all minute details.
- (b) The sūtras related to the compound formation are of considerable and manageable size (around 400 in number).
- (c) While most of these sūtras are vidhi sūtras, among them there are niṣedha and adhikāra sūtras as well.

The main goal of our research is to generate Sanskrit compounds automatically from a given analytical paraphrase. This process starts with an

alaukikavigraha (analytical paraphrase) and undergoes a series of transformations, resulting into a compound. Then the question rises: up to what extent can the compounds be generated automatically? For that we decided to build a computational model for compound generation, and study those cases where the machine needs some extra linguistic information for processing.

The organisation of thesis

In the first chapter we survey the literature of earlier works with focus on the compounds and computational treatment of Aṣṭādhyāyī, and state the goal of my research work.

As we are aware Pāṇini does not describe the compound formation procedure as such at one place. But later grammarians examined the rules and came up with an order of the operations to form a compound. In the second chapter we describe various steps involved in the entire compound formation following Pāṇinian tradition.

In the third chapter we discuss the issues related to the order of various steps involved in compound formation, especially with reference to the deletion of suffixes and addition of samāsānta suffix. We discuss what the commentators like Patañjali, Nageśa and Dīkṣita have to say on these issues, and explain with examples why Patañjali's treatment is better.

In the fourth chapter, we give the classification of all compound related sūtras based on the type of information each sūtra needs. We noted down the kind of information each sūtra needs for implementation. We observe that in some cases this information is just phonetic or morphological. But there are some sūtras, though less, which need semantic information of the lexical

items such as the meaning, the class property, its relation to other word, and so on. In some small number of cases even pragmatic information such as whether the compound is being used to indicate a particular sense or not is needed. This study helps in developing the strategy for computational modeling of the sūtras, and the organisation of lexicon.

In the fifth chapter we study various semantic information needed for the compound formation, and the aim of this study is to arrive at a structured lexicon with all the needed semantic features associated with it. We look at the Amarakośa, and the Nyāya - Vaiśeṣika ontology, and the earlier work of Nair (2011) who enriched Amarakośa with the Nyāya - Vaiśeṣika ontology, and suggest some ammendments in the structure in order to make it useful for our task.

The sixth chapter gives a detail description of the implementation of the compound generation module wise. We have implemented the compound generator using the lexical analyser ‘Lex’ and a parser generator ‘Yacc’. Some of the modules were written as rewrite rules in Yacc, and most of the others are implemented using a lexer. Each module implemented with a lexer is thus a finite state automaton. We have discussed the cases where we had to deviate from Pāṇini’s representation in order to ‘transcode’ them in programming environments of Lex and Yacc. In some cases we had to split a rule into many, in some other cases the rules were better represented as a function than just pattern matching.

The seventh chapter contains the conclusion part in which we have discussed the limitations and future directions for improving the Sanskrit compound generator.

This implementation has provided us clues for enriching the lexicon. The lexical analyser is sufficient to model the computational complexity of the rules. In other words the rules are so simple that the power of regular expressions is sufficient in most of the cases. We have tested this generator against the examples from Pāṇinīyavyākaraṇodāharaṇakośaḥ.

The second part of the dissertation contains complete code of the implementation.

Chapter 2

Compound Formation : Pāṇinian Procedure

Compounds being an important part of the language, Pāṇini deals with them very thoroughly handling the exceptional cases in great detail. Out of around 4000 sūtras in Aṣṭādhyāyī a little more than 400 sūtras deal with the compound formation. In the first chapter of Aṣṭādhyāyī sūtras from 1.4.61 to 1.4.79 are related to the gati samāsa. Sūtrās starting from A2.1.1 to A2.2.29 are related to the formation of a compound (*vidhāyaka sūtras*). Sūtrās from A2.2.30 to A2.2.38 are related to the change of word order (*pūrvanipāta*) in the compound formation process. In the 5th chapter sūtras from A5.4.68 to A5.4.160 introduce special suffixes related to the compounds called *samāsānta pratyayas*. The 3rd section of the 6th chapter has several sūtras that deal with the various stages of compound formation. Sūtrās from A6.3.1 to A6.3.24 prohibit the deletion of the case suffix, The sūtras which transform the first constituent and second constituent of a compound start from 1st and 3rd section of the 6th chapter.

The most relevant paper for my work is by Mahavir (1986). In this paper he discusses the treatment of compounds in Pāṇini. In the first section he discusses the mutual compatibility (sāmarthya) between the constituents of the compound. He gives the justification for the order in which Pāṇini deals with different compound types. He observes that if Avyayībhāva class were placed after Tatpuruṣa or Bahuvrīhi then it would cease to be an exception of either Tatpuruṣa or Bahuvrīhi and would result in the generation of non-existent irregular forms.

In the second section he talks about the various transformations a vigra-havākya undergoes resulting into a compound form.

These transformations respond to various operations such as

- (a) “Fixation of the order of the constituents.
- (b) Assignment of the prātipadika (nominal stem) designation to this string.
- (c) Morpho-phonemic changes in pūrvapada and uttarapada.
- (d) Change of gender and number in certain types, and
- (e) Change or retainment of original accent”.

We describe below the steps involved in compound formation process followed in the Pāṇinian tradition.

1. ‘Analytical Paraphrase’ (*Alaukikavigraha*)

Assume that a speaker wants to express the concept ‘a person who is a servant of a king’. Then he has three basic conceptual elements viz. a person, a king and the servant-master relation between them.

Now this servant-master relation *sevyā-sevaka-bhāva* in Sanskrit is expressed by *ṣaṣṭhī vibhakti* (6th case suffix). So the concept is represented in the form of a linguistic expression as

$$rājan + \dot{n}as\ puruṣa + su$$

where the words *rājan* and *puruṣa* denote the objects ‘a king’ and ‘a person’, and the suffix *nas* represents the relation between these two elements, and ‘su’ is the singular nominal case suffix. This representation is called an *alaukikavigraha* and is the starting point of the derivation of a compound. This *alaukikavigraha* is the paraphrase of the compound showing the linguistic analysis and thus is the analytic paraphrase of the compound.

2. (a) *Samāsa* designation: In the above example, since the components *rājan* and *puruṣa* have mutual meaning compatibility, if the speaker has an intention to form a compound, Pāṇinian sūtra *ṣaṣṭhī P2.2.8* sanctions the formation of such a compound. Such an *alaukikavigraha* then gets a **Samāsa** designation. The compound formation is purely under the dictum of the desire of a speaker (*vaktṛvivakṣādhīnam*). In the derivation process, corresponding to an *alaukikavigraha* the speaker has an option to produce either a sentential expression or a compound expression.¹ Of course, only if the language and hence the grammar allows such formation, a speaker can form a compound. The choice is governed by a condition that a compound may be formed only if the

¹Strictly speaking, there are certain compounds which are termed as *nityasamāsa* (*avigrahaḥ asvapadavigraho vā nityasamāsaḥ*) which do not have a sentential paraphrase. Such compounds are governed by the sūtras falling in the range 2.1.5 to 2.1.10. Barring these, all the compounds can optionally be expressed as a sentential paraphrase as well.

components involved are mutually compatible.² In other words, there should be some rule in the grammar, which guarantees the formation of a compound with given components and meanings.

(b) Introduction of *samāsānta* *pratyayas*:

There are certain suffixes which are added to the compound at the end. These are called *samāsānta pratyayas*. These are added at the end of an *alaukikavigraha* and are treated as a part of a compound. For example, the *sūtra avyayībhāve śaratprabhṛtibhyaḥ* (A5.4.107) adds a *ṭac* suffix if the compound is of type *avyayībhāva* and one of the word from the list starts with *śarad*. Thus

$$\textit{śarad} + \textit{ṇas upa} \rightarrow \textit{śarad} + \textit{ṇas upa tac}.$$

3. Declaring the string to be a nominal stem (*prātipadikasañjñā*)

At this point, now the stage is set to generate a new nominal stem (*prātipadikam*) from this expression and thus the whole expression is designated with a label *prātipadikam*³.

4. Deciding the word order (*upasarjanasañjñā*)

The head (*viśeṣya*) of a compound more-or-less depends on the type of a compound. The *Tatpuruṣa* (endocentric) compound has second component as the head while the *Avyayībhāva* has the first component as the head. In case of *Bahuvrīhi* (exo-centric) the head is the one which this Bahuvrīhi compound modifies and in case of conjunctive compounds, both the words together are heads. However there are many exceptions to this general observation. So the semantic criterion

²samarthaḥ padavidhiḥ (A2.1.1)

³kṛttaddhitasamāsāśca (A1.2.46)

for head does not help us in deciding the word order. Pāṇini uses a special mechanism to mark the order of words in a compound. In Pāṇini's grammar he labels the component which occupies the first place by the term *upasarjana*. The designator of a word in nominative case in the rule that decides the compound type is the one which gets the designation *upasarjana*. For example consider the input

śarad+ nas upa

The sūtra⁴ which sanctions compound formation for this input consists of two words. Among these the word *avyayam* in the sūtra is in nominative. Hence that word in the *alaukikavighraha* which is an *avyaya* gets a designation *upasarjana*. In the above example *upa* is an *avyaya*, and therefore *upa* gets a designation *upasarjana*.

5. Change of word order (pūrvanipātaḥ)

The consequence of assigning a label *upasarjana* is to place the word having designated as *upasarjana* in the 1st place. This action is termed as *pūrvanipātaḥ*. In the above example, 'upa' which got *upasarjanasāñjñā* occupies the first place. This changes the order of the input string to

upa śarad + nas.

6. Deletion and Non-deletion of case suffixes (Subluk and aluk kārya)

There are certain exceptional cases, when the case suffix of the involved words does not get elided⁵. For example, in the following case

yudh +ni ⁶ *sthira + su*

ni is retained. But in the case of

⁴*avyayam vibhaktisamīpasamṛddhivyṛddhyarthābhāvātyayāsampratiśabdaprādurbhāva-paścādyathānupūrvyayaugapadyasādrśyasampattisākalyāntavacanēsu* (A2.1.6)

⁵*aluguttarapade* (A6.3.1)

⁶*haladantatsaptamyāḥ sañjñāyām* (A6.3.9)

rājan+ñas puruṣa+su

ñas is deleted. In either case the case suffix of the second word is deleted, This will result the above cases into
yudh +ñi sthira and *rājan puruṣa*.

7. Transformation of the first component

In certain cases, the first component undergoes certain transformations in the presence of the second component. For example, *pāda* changes to *pad* if it is followed by *āti*.

pāda + āti → pad + āti

8. Transformation of the second component

In certain cases, the second component also undergoes certain changes in the presence of the first component. For example, *sthira* changes to *ṣṭhira* in the presence of *yudhi*.

yudhi + sthira → yudhi ṣṭhira

9. Padakārya

Each of the components thus transformed obligatorily undergo certain operations which are independent of the other words in the context. For example, in the case of *rājan* ‘n’ gets deleted.⁷

10. Joining of the words (sandhikārya)

The two components thus transformed, now are joined together with a sandhi operation. For example,

rāma ālaya → rāmālaya

11. Deciding the gender of a derived nominal stem (liṅganirdhāraṇam)

Such a stem is now all set to undergo word formation rules. But before that, this stem should get the proper gender. Gender of a nominal lex-

⁷*nalopaḥ prātipadikāntasya* (A8.2.7)

icon is the property of the lexical item⁸. An *Avyayībhāva* compound is always in neuter gender⁹. While in the case of *Tatpuruṣa* (endocentric) and *Dvandva* (conjunctive) compounds¹⁰ the gender of the compound is same as the second component. In the case of *Bahuvrīhi* (exocentric), the gender is same as that of the word it modifies. There are of course a few exceptional cases which are dealt with special rules by Pāṇini.

12. Deciding the number

The number of the resulting compound depends on its constituents. In the case of *Tatpuruṣa* the number depends on the primary component and in *Bahuvrīhi* it depends on the referent of the resulting compound. *Avyayībhāva* compound has singular number and in the case of *iterataradvandva*, the number of the resulting compound will be dual with some exceptions where the number may be plural. In *saṁāhāradvandva* the number will be singular.

13. Deciding the svāra (accent)

A compound is characterized by a single accent. Usually the final vowel in a compound is a *udātta* (high pitch). *Tatpuruṣa* compounds have a high pitch at the end (*antodātta*) and *Bahuvrīhi*, *Avyayībhāva* and *Dvandva* compounds will have the same accent as that of the *pūrvapada* (*pūrvapada*prākṛti svāra). These are the general rules and there are some exceptions to them.

14. Word Formation

The final step is the word formation, with appropriate vibhakti.

⁸*prātipadikārtha liṅgaparimāṇavacanamātre prathamā* (A2.3.46)

⁹*avyayībhāvaśca* (A2.4.18)

¹⁰*paravalliṅgam dvandvatatpuruṣayoḥ* (A2.4.26)

Chapter 3

Theoretical Issues in Compound Formation

In the previous chapter we saw various steps involved in the process of compound formation. There are differences of opinion regarding the order in which these steps are to be carried out. In this chapter, we discuss the issues related to the order of various steps involved, and the consequences thereof.

Typical process of compound formation followed by the teachers is sketched below. Pāṇinīyavyākaraṇodāharaṇakośa shows the process for thousands of compounds. The generation process of the compound starts with alaukikav-igraha, and it gets the designation ‘samāsa’ followed by another designation ‘prātipadika’. Then aluk related operations are performed. This is followed by the elision of suffixes. Then the word order related operations viz. designation of upasarjanasañjñā followed by pūrvanipāta take place. Additional suffix called samāsāntā suffix is added at this point. This is then followed

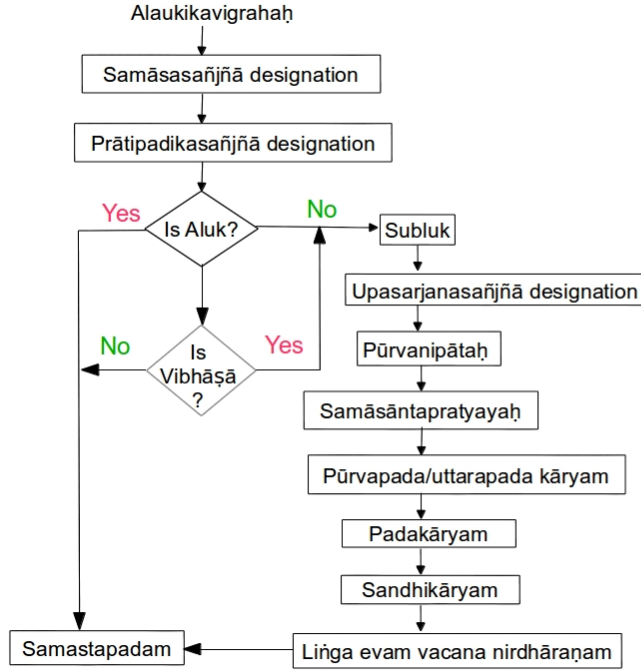


Figure 3.1: Process of Compound Generation (1)

by pūrvapadakārya, uttarapadakārya, sandhikārya and finally the gender is decided and appropriate vibhakti is added to get the compound word. This process is shown as a flow chart in Figure 3.1

There are differences of opinion with this order, especially with reference to the deletion of suffixes and addition of samāsānta suffix.

- When should vibhakti suffix (sup) be elided?
Should it be elided before the designation of upasarjanasañjñā or after?

- At what stage should the samāsānta suffix be added ?

Should we add this suffix at the starting point of the samāsa prakriyā or after deciding the word order of the components ?

Let us see what the commentators have to say on the samāsānta suffix and the arguments in favour and against the deletion of nominal suffix (subluk) before and after the assignment of upasarjanasañjñā.

3.1 When should subluk take place?

If one applies the subluk before assigning the upasarjanasañjñā then it creates a problem. The case suffix is the basis of information for deciding upasarjanasañjñā. Hence once this is deleted, there is no way to decide which of the components will get an upasarjanasañjñā.

We illustrate this with an example. Consider the alaukikavighraha rājan + ñas puruṣa + su. The word rājan + ñas gets the designation upasarjana. The governing sūtra is ṣaṣṭhī (A2.2.8). The sūtra prathamā nirdiṣṭam samāsa upasarjanam (A1.2.43) says that in the samāsavidhāyakasūtra (a sūtra which dictates the type of the resulting compound), the word which is in nominative case will get designated as upasarjana. Now in the sūtra ṣaṣṭhī (A2.2.8), there is only one word, and it is in nominative. Hence the word in genitive case in the alaukikavighraha viz. rājan gets designated as upasarjana.

Now if the case suffix is deleted first, then in the absence of the genitive suffix, it is not possible to designate rājan an upasarjanasañjñā. This is the objection. In response, some argue that with the help of a sūtra pratyayalope pratyayalakṣaṇam (A1.1.62)(an operation conditioned by an affix applies

even if the affix has been replaced by zero¹), we can designate it as upasarjana even after the subluk. There is no bādhyabādhakabhāva between the two operations - subluk and upasarjana.

However, as is evident from this discussion even if there is no bādhyabādhakabhāva, in order to designate upasarjanasañjñā after the deletion of case suffixes, one needs to invoke another sūtra pratyayalope pratyayalakṣaṇam (A1.1.62). This is a gaurava (prolixity). If we designate the word with an upasarjanasañjñā based on the case suffix and then we delete it, we need not invoke pratyayalope pratyayalakṣaṇam (A1.1.62). This brings in lāghava (brevity).

3.2 Meaning of the sūtra samāsāntāḥ (A5.4.68)

Pāṇini has introduced 21 suffixes called samāsānta suffixes in the Aṣṭādhyāyī from 5.4.68 to 5.4.160. They are: (1) ḍac (2) a (3) ac (4) aḥnādeśa (5) ṭac (6) ṣac (7) ṣa (8) ap (9) ac (10) asic (11) anic (12) ic (13) jñu (14) anan (15) niṇ (16) it (17) lopa (18) datṛ (19) lopa (20) hṛt (21) kap

The sūtra samāsāntāḥ (A5.4.68) says that “The tadditha affix introduced, henceforth, occur at the end (anta) of a nominal stem termed compound (samāsa)”². Patañjali interprets the word samāsa in the sūtra samāsāntāḥ (A5.4.68) as an analytical paraphrase (alaukikavigrahaḥ) and Bhaṭṭoji Dīkṣita interprets the word samāsa as an analytical paraphrase as well as an uttarapada.

¹The Aṣṭādhyāyī of Pāṇini Vol. II Ramanath Sharma, Page No.63, 2000

²The Aṣṭādhyāyī of Pāṇini Vol. IV Ramanath Sharma, Page No.704, 2000

3.3 At what stage should the samāsānta suffix be added?

We present below the interpretation offered by Bhaṭṭoji, followed by Patañjali and Nāgeśa Bhaṭṭa.

Bhaṭṭoji Dīkṣita in his book Praudhamanoramā states that the samāsānta suffix can become a part of either the analytical paraphrase or an uttarapada. According to him, there are cases in which one needs to treat the samāsānta suffix as a part of the complete paraphrase and cases where one needs to consider the samāsānta suffix as a part of an uttarapada. We explain both these cases with an example each.

Example 1:

alaukikavigrahaḥ : upa + su śarad + ŋas

samāsānta pratyayāḥ : [upa + su śarad + ŋas [t̪]a[c]]

where [t̪] and [c] are ‘it’ markers, and the whole expression gets the samāsasañjñā.

prātipadikasañjñā : ‘upa + su śarad + ŋas a’ , now it gets prātipadikasañjñā

upasarjanasañjñā : **upa** + su śarad + ŋas a, upa gets upasarjanasañjñā

pūrvanipātaḥ : upa + su śarad + ŋas a

subluk : upa śarad a

sandhikāryam : upaśarada

liṅganirdhāraṇam : upaśarada , neuter gender

vibhaktikāryam : upaśarada + su → upaśaradam, due to the replacement of ‘a’ by ‘am’.

Now instead, if the samāsānta suffix is added to the uttarapada the situation will be

alaukikavigrahaḥ : [upa + su śarad + ṇas]

samāsānta pratyayāḥ : [upa + su (śarad + ṇas) [ṭ]a[c]] where [ṭ] and [c] are ‘it’ markers. upa + su śarad + ṇas gets samāsasañjñā and ṭac is not part of the samāsasañjñā. But śarad + ṇas ṭac is an uttarapada. We have marked the expression that gets samāsasañjñā by [] and an expression in () is the uttarapada. Note here that ‘a’ of the samāsānta suffix is now part of the uttarapada but not of the samāsa.

prātipadikasañjñā : [upa + su (śarad + ṇas) a]

upasarjanasañjñā : [upa + su (śarad + ṇas) a]

pūrvanipātaḥ : [upa + su (śarad + ṇas) a]

subluk : [upa (śarad) a]

sandhikāryam : [upa(śarad) a]

liṅganirdhāraṇam : [upa (śarad) a], neuter gender

vibhaktikāryam : [upa(śarad) a]

Thus here ‘a’ of [ṭ]a[c] is part of uttarapada but is not a part of the samāsa. And hence the replacement of ‘a’ by ‘am’ is prohibited.

Now we look at another example, which Bhaṭṭoji Dikṣita used to show the necessity of considering samāsānta suffix as a part of uttarapada.

Example 2:

alaukikavigrahaḥ : [dvi + os pur + os]

samāsānta pratyayāḥ : [dvi + os (pur + os) a], As in the previous case,

we mark the boundaries of samāsasañjñā by [] and of uttarapada by ().

prātipadikasañjñā : [dvi + os (pur + os] a)

upasarjanasañjñā : [dvi + os (pur + os] a)

pūrvanipātaḥ : [dvi + os (pur + os]a)

subluk : [dvi (pur] a)

sandhikāryam :[dvi(pur] a)

liṅganirdhāraṇam : [dvi(pur] a), feminine gender

vibhaktikāryam : [dvi(pur] a)→ dvipura + nīp → dvipurī.

In this example the samāsānta suffix ‘a’ is a part of an uttarapada. Hence by the vārtika ‘akārāntottarapado dviguḥ striyām bhāṣyataḥ iti vaktavyam’ the feminine suffix will be added to the entire compound. The vārtika says that in a dvigu compound if the uttarapada is a prātipadika ending in ‘a’, then the feminine suffix nīp is added to the compound. On the other hand if we add the samāsānta suffix to the entire compound then the uttarapada ‘pur’ does not end in ‘a’. So there is no scope for the addition of feminine suffix.

From these two examples, it is obvious that in one example, to arrive at the desired form one needs to accept that the samāsānta suffix as a part of the alaukikavigraha while in the other, as a part of the uttarapada. But if we look at Patañjali’s commentary, then it is obvious that he subscribes to only the first view viz. that the samāsānta suffix is a part of complete alaukikavigraha, and not the uttarapada.

The discussion on the status of samāsānta pratyayas in Mahābhāṣya appears under the commentary of the sūtra ‘gostriyorupasarjanasya’ (A1.2.48).

In the case of a compound formation ‘bahu + jas kumārī + jas’ there is a scope to introduce a samāsānta suffix ‘kap’. There is also a possibility of shortening of the vowel ī in kumārī. In the desired form there is no shortening of the vowel ī. So Patañjali discusses which sūtra prohibits the shortening of the vowel ī, and why.

First Patañjali rules out the possibility that the shortening of the vowel ī is prohibited due to the vārtika ‘kapi ca’. In fact he questions the very necessity of this vārtika and claims that this vārtika is not needed and the sūtra ‘na kapi’ (A7.4.14) should operate. In such cases the objection that ‘na kapi’ (A7.4.14) is applicable only to prohibit the results of ‘ke’ṇaḥ’ (A7.4.13) is ruled out by him and he interprets ‘na kapi’ (A7.4.14) to prohibit vowel shortenings in all the cases in the presence of a ‘kap’ suffix.

Now the question is when both the vowel shortening as well as introduction of samāsānta suffix ‘kap’ are possible, which operation takes place first? The one with less number of conditions is treated as antaraṅga and the other one is bahiraṅga. The operation of vowel shortening requires the prātipadikam to be feminine. Whereas the addition of samāsānta suffix ‘kap’ takes place provided the compound is of type Bahuvrīhi and the second component of compound ends in either ‘ī’ or ‘ū’ (nadī). Thus the number of pre-requisites for vowel shortening are less compare to the process of addition of samāsānta suffix ‘kap’. Hence vowel shortening is treated as an antarṅgakārya and addition of samāsānta suffix ‘kap’ is a bahiraṅgakārya, (alpāpekṣamantaraṅgam bahvapekṣam bahiraṅgam) based on the paribhāṣā *asiddham bahiraṅgamantaraṅge* it appears that vowel shortening takes place first. But Patañjali calls addition of ‘kap’

as an antaraṅgataṛaḥ, and hence the samāsānta suffix is added first, and then by the sūtra ‘na kapi’ (A7.4.14) vowel shortening will not take place. Patañjali further says that the words that are to be compounded will wait for the samāsānta suffix because it is a part of the compound. Hence this suffix ‘kap’, in the example ‘bahukumārīkaḥ’, is a part of the analytical paraphrase so it is an antaraṅgataṛaḥ and vowel shortening is bahiraṅga. Therefore vowel shortening operation doesn’t take place.

Nageśa reinforces the interpretation of Bhāṣyakāra Patañjali that in sūtras ‘samāsāntāḥ’ (A5.4.68) and ‘antaḥ’ (A6.2.92), the samāsānta suffix is added at the end of an analytical paraphrase and becomes a part of the compound and then the entire string gets the designation samāsa.

So, based on the above discussions we can say that the samāsānta suffix is a part of an analytical paraphrase (alaukikavigraha) only, and the alaukikavigraha along with the samāsānta suffix gets the designation samāsa. Dīkṣita’s view on samāsānta suffix as a part of an uttarapada is not acceptable. The reason is there is no such evidence (pramāṇa) that tells us the word samāsa in samāsānta as a uttarapada. Dīkṣita has extended the meaning of the word samāsa in samāsānta as a uttarapada by lakṣaṇa to explain the correct forms such as dvīpurī, dvīdhurī. But Bhāṣyakāra does not give this explanation. Then the question may arise how does he handle the cases like dvīpurī etc. Patañjali rejects the vārtika ‘akārāntottarapado dvīguḥ striyām bhāṣyataḥ iti vaktavyam’, because in the sūtra ‘dvīguḥ’ (A4.1.21) he says that the feminine suffix is added to the entire dvīgu compound which ends in ‘a’. So, there is no need to add the ‘a’ suffix to uttarapada based on the vārtika. Therefore, the samāsānta suffix becomes a

part of the analytical paraphrase only.

In the implementation of the compounds we follow the Bhāṣyakāra's interpretation i.e the samāsānta suffix is a part of the analytical paraphrase.

Now the generation process of the compounds which is inferred from the traditional commentators like Patañjali, Nageśa etc, starts with the alaukikavigraha, and then samāsāntā suffix is added (additional suffixes), and it gets the designation samāsa, such a samāsa gets the designation prātipadikasañjñā after that aluk (in the case of aluk compounds), upasarjanasañjñā, pūrvanipāta, subluk, transformation of the components (pūrvapadakārya and uttarapadakārya, sandhikārya, liṅganirdhāraṇam and finally vibhaktikāryam) are carried out to get the compound word.

Figure 3.2 shows the process of the generation of the compounds, implemented by us.

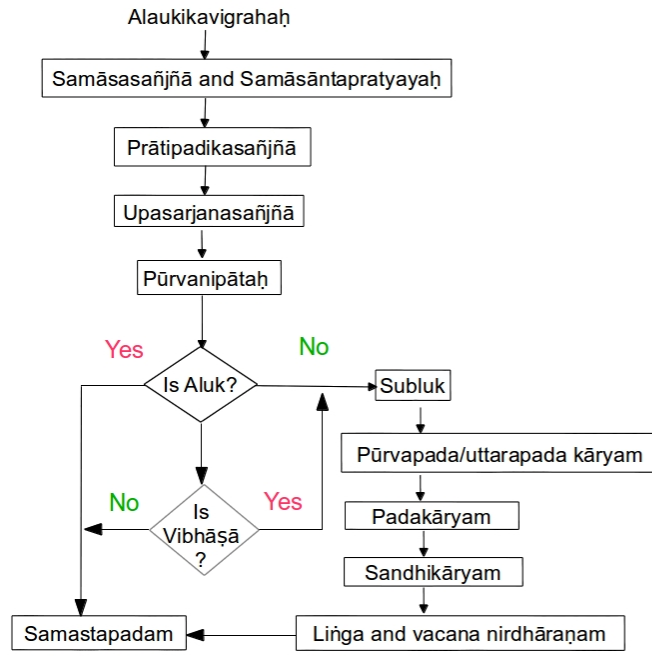


Figure 3.2: Process of Compound Generation (2)

Chapter 4

Analysis and Classification of Sūtras

Aṣṭādhyāyī has around 400 sūtras that deal with the compound formation. As we noticed in previous chapters, the compound formation process is completely algorithmic. So the questions that arose in my mind were

1. Is it possible to automate the process of compound formation?
2. If yes, how many sūtras are completely formal, and how many of them can't be automated?
3. In case of sūtras which can't be automated, what kind of information is needed for processing?

In order to answer these questions, we classified the sūtras based on the operation they correspond to in the compound formation. Then, we looked at each sūtra and analysed them to find out what kind of information is needed by the sūtra for its application.

While classifying the sūtras according to their function, we observed

the following:

1. Most of the sūtras are conditional expressions specifying certain action if a certain condition is satisfied. For example,:

pādasya padājyātigopahateṣu A6.3.52

This sūtra says if the first constituent is *pāda* and the second constituent is either *āti*, *ga* or *upahata* then *pāda* changes to *pad*.

2. There are some sūtras which require a rich structure involving deeper analysis than just the sequence of phonemes. we could classify them in four types.

(a) The morphological analysis of a word

Some sūtras require morphological information of components, for example whether the component is a word in a feminine gender with a suffix *ñīp*, or whether it is a primary derivative with a ‘*kta*’ or ‘*lyap*’ and so on. For example, the rule ‘*svayaṃ ktena*’ A2.1.25 says that if the first word is ‘*svayaṃ*’ and the second word ends in a primary derivative suffix ‘*kta*’ then a compound may be formed.

(b) The semantic category of a word

Many sūtras need semantic information to form a compound. For example, The sūtra ‘*ktenāhorātrāvayavāḥ*’ A2.1.45 says that if the first constituent is a word denoting part of either day or night and if it is followed by a word with primary derivative suffix ‘*kta*’ then a compound may be formed. Here the information that the word is a part of day/night is semantic in nature.

(c) Some sūtras require pragmatic information. For example, the sūtra

‘ṣaṣṭyā ākroṣe’ A6.3.21 says that in a ṣaṣṭhī tatpuruṣa compound if a word which is in genitive case indicates the sense ākroṣa (censure) then the non-elision of case suffix is ordained. As is evident, the information whether the word is being used with certain emotions can not even be associated with a lexicon and needs to be obtained only through the context. Hence this type of information is pragmatic by nature.

- (d) Finally Pāṇini has listed a set of compounds which are exceptional cases. There are a few of such lists such as ‘tiṣṭadguprabṛūni ca’ A2.1.17, ‘mayūravyaṃsakādayaśca’ A2.1.72 etc. These lists are called gaṇapāṭhas.

Thus we observed that for computational implementation of the sūtras, we need

- A list of gaṇapāṭhas,
- A lexicon tagged with semantic information, and
- A morphological analyser to analyse the components.

The sūtras which need pragmatic information are difficult to process automatically.

We have classified all the compound related sūtras based on the type of information each sūtra needs. Following tables list all the sūtras, grouped under the operation they correspond to. Against each sūtra we have noted down what kind of information each sūtra needs.

In this chapter we discuss the semantic and pragmatic information needed for the implementation.

samāsavidhāyakakārya	
Input needed for implementation	sutras
gaṇapāṭhaḥ	<i>tiṣṭhadguprabhṛtīni ca</i> 2.1.16
	<i>yājakādibhiśca</i> 2.2.9
	<i>saptamīśauṇḍaiḥ</i> 2.1.40
	<i>pātresamitādayaśca</i> 2.1.48
	<i>upamitaṃ vyāghrādibhiḥ</i> <i>sāmānyāprayoge</i> 2.1.56
	<i>śreṇyādayaḥ kṛtādibhiḥ</i> 2.1.59
	<i>kumāra śramaṇādibhiḥ</i> 2.1.70
	<i>mayūravyaṃsakādayaśca</i> 2.1.72
	<i>sākṣātprabhṛtīni ca</i> 1.4.74
	<i>ūryādicvidācaśca</i> 1.4.60
Semantic category	
saṃkhyā	<i>akṣaśalākāsaṃkhyāḥ pariṇā</i> 2.1.10
	<i>saṃkhyā vaṃśyena</i> 2.1.18
	<i>saṃkhyāpūrvō dviguḥ</i> 2.1.51
guṇavacana	<i>tr̥tīyā tatkr̥tārthena guṇavacanena</i> 2.1.29
naḍivācakāḥ	<i>naḍībhiśca</i> 2.1.19
vyañjana	<i>annena vyañjanam</i> 2.1.33
bhakṣya	<i>bhakṣyeṇa miśrīkaraṇam</i> 2.1.34
dūrārtha	<i>stokāntikadūrārthakṛcchrāṇi ktena</i> 2.1.38

pūraṇa/guṇa/avyaya	<i>pūraṇa-guṇa-suhitārtha-sadavyaya-tavya-samānādhikaraṇena</i> 2.2.11
parimāṇa	<i>kālāḥ parimāṇinā</i> 2.2.5
sañjñā	<i>sañjñāyām</i> 2.1.44
	<i>anyapadārthe ca sañjñāyām</i> 2.1.20
ahorātrāvayava	<i>ktenāhorātrāvayavāḥ</i> 2.1.45
dik/saṅkhyā/sañjñā	<i>diksaṅkhye sañjñāyām</i> 2.1.50
jāti	<i>katarakatamau jātiparipraśne</i> 2.1.63
	<i>poṭāyuvati</i> 2.1.65
	<i>kr̥tyatulyākhyā ajātyā</i> 2.1.68
praśaṃsāvacana	<i>praśaṃsāvacanaīśca</i> 2.1.66
varṇa	<i>varṇo varṇena</i> 2.1.69
catuspād	<i>catuspādo garbhīṇyā</i> 2.1.71
saṅkhyā evam avyayā	<i>saṅkhyāvyayāsanna-dūrādhika-saṅkhyāḥ saṅkhyeye</i> 2.2.25
dik	<i>diñnāmānyantarāle</i> 2.2.26
prāṇi/tūrya/senāṅga	<i>dvandvaśca</i> <i>prāṇitūryasenāṅgānām</i> 2.4.2
caraṇa	<i>anuvāde caraṇānām</i> 2.4.3
jati/aprāṇi	<i>jātiraprāṇiṇām</i> 2.4.6
nadī/deśa/grāmāḥ	<i>viśiṣṭaliṅgo nadīdeśo'grāmāḥ</i> 2.4.7
kṣudrajantava	<i>kṣudrajantavaḥ</i> 2.4.8
adhvaryukratuḥ	<i>adhvaryukraturanapumsakam</i> 2.4.4

vṛkṣa/mṛga/tṛṇa/dhānya/vyañjana	<i>vibhāṣā vṛkṣa-mṛga-tṛṇa-dhānya-vyañjana-paśuśakunyaśvavadava-pūrvāparādhharottarāṇām</i> 2.4.12
āyāmaḥ	<i>yasya cāyāmaḥ</i> 2.1.15
sādrśya	<i>yathā'sādrśye</i> 2.1.7
kṣepa	<i>khaṭvā kṣepe</i> 2.1.25
	<i>kṣepe</i> 2.1.47
	<i>kiṃ kṣepe</i> 2.1.64
atyantasam̐yoga	<i>atyantasam̐yoge ca</i> 2.1.28
adhikāṛthavacana	<i>kṛtyairadhikāṛthavacane</i> 2.1.32
ṛṇa	<i>kṛtyairṛṇe</i> 2.1.43
upamāna	<i>upamānāni sāmānyavacanaiḥ</i> 2.1.55
pūjā	<i>ktena ca pūjāyām</i> 2.2.12
	<i>sanmahatparamottamotkr̥ṣṭāḥ</i> <i>pūjyamānaiḥ</i> 2.1.61
	<i>vṛndārakanāgakuñjaraiḥ</i> <i>pūjyamānam</i> 2.1.62
adhikaraṇavācī	<i>adhikaraṇavācinā ca</i> 2.2.13
	<i>vipratīṣiddham cānadhikaraṇavāci</i> 2.4.13
ekadeśī	<i>pūrva-parādhharottaram-</i> <i>ekadeśinaikādhikaraṇe</i> 2.2.1

Morph analysis is needed	
karma	<i>karmaṇi ca 2.2.14</i>
kartari	<i>tr̥jakābhyāṃ kartari 2.2.15</i>
	<i>kartari ca 2.2.16</i>
No extra information needed for implementation	
<i>saha supā 2.1.4</i>	
<i>svayaṃ ktena 2.1.24</i>	
<i>sāmi 2.1.26</i>	
<i>kālāḥ 2.1.27</i>	
<i>karṭṛkaraṇe kṛtā bahulam 2.1.31</i>	
<i>pañcamī bhayena 2.1.36</i>	
<i>apetāpoḍha-mukta-patitāpatrastairalpaśaḥ 2.1.37</i>	
<i>ṣaṣṭhī 2.2.8</i>	
<i>nityaṃ kr̥dājīvikayoḥ 2.2.17</i>	
<i>na nirdhāraṇe 2.2.10</i>	
<i>ardhaṃ napuṃsakam 2.2.2</i>	
<i>dvitīya-tr̥tīya-caturtha-turyāṇyanyatarasyām 2.2.3</i>	
<i>prāptāpanne ca dvitīyayā 2.2.4</i>	
<i>siddhaśuṣkapakvabandhaiśca 2.1.41</i>	
<i>dhvāṅkṣeṇa kṣepe 2.1.42</i>	
<i>tatra 2.1.46</i>	
<i>ktena nañviśiṣṭenānañ 2.1.60</i>	
<i>yuvākhalatipalitavalinajaratībhiḥ 2.1.67</i>	
<i>kugatiprādayaḥ 2.2.18</i>	
<i>tatropapadaṃ saptamīstham 3.1.92</i>	

<i>upapadamatiṅ 2.2.19</i>
<i>amaivāvyayena 2.2.20</i>
<i>tr̥t̥yāprabhṛtīnyanyatarasyām 2.2.21</i>
<i>ktvā ca 2.2.22</i>
<i>anekamanyapadārthe 2.2.24</i>
<i>tatra tenedamiti sarūpe 2.2.27</i>
<i>tena saheti tulyayoge 2.2.28</i>
<i>cārthe dvandvaḥ 2.2.29</i>
<i>yeṣāṃ ca virodhaḥ śāśvatikaḥ 2.4.9</i>
<i>taddhitārthottarapadasamāhāre ca 2.1.52</i>
<i>kutsitāni kutsanaiḥ 2.1.53</i>
<i>pāpāṇake kutsitaiḥ 2.1.54</i>
<i>īṣadakṛtā 2.2.7</i>
<i>lakṣaṇenābhipratī ābhimukhye 2.1.13</i>
<i>yāvadavadhāraṇe 2.1.8</i>
<i>suppratinā mātrārthe 2.1.9</i>
<i>anuryatsamayā 2.1.14</i>
<i>adhyayanato'viprakṣṭākhyāyām 2.4.5</i>

samāsāntāḥ	
Input needed for implementation	sutras
gaṇapāṭhaḥ	avyayībhāve śaratprabhṛtibhyaḥ 5.4.107
	pādasyalopo'hastyādibhyaḥ 5.4.138
	kumbhapadīṣu ca 5.4.139
	dvidanḍyādibhyaśca 5.4.128
	uraḥ prabhṛtibhyaḥ kap 5.4.151
Semantic category	
svāṅgavācī	nāḍītantryoḥ svāṅge 5.4.159
	bahuvrīhau sakthyakṣṇoḥ svāṅgātṣac 5.4.113
prāṇivācī	upamānādaprāṇīṣu 5.4.97
janapadavācī	brahmaṇo janapadākhyāyām 5.4.104
saṅkhyeyavācī	bahuvrīhau saṅkhyeye ḍajabahugaṇāt 5.4.73
sañjñāvācī	aññāsikāyāḥ sañjñāyām nasam cāsthūlāt 5.4.118
saṅkhyāvācī	tatpuruṣasyāṅguleḥ saṅkhyāvyayādeḥ 5.4.86
kṣepavācī	kimaḥ kṣepe 5.4.70
jāti evam sañjñāvācī	ano'smāyassarasām jātisamjñayoh 5.4.94

sañjñāvācī	<i>vā sañjñāyām</i> 5.4.133
	<i>na sañjñāyām</i> 5.4.155
	<i>agrākhyāyāmurasah</i> 5.4.93
	<i>striyām sañjñāyām</i> 5.4.143
karmavyatihārah	<i>ickarmavyatihāre</i> 5.4.127
āyāmavācī	<i>anugavamāyāme</i> 5.4.83
upamāna	<i>upamānācca</i> 5.4.137
sañkhyā	<i>sañkhyā supūrvasya</i> 5.4.140
Pragmatics	
pūjā	<i>na pūjanāt</i> 5.4.69
adarśanam	<i>akṣṇo'darśanāt</i> 5.4.76
vandanam	<i>vandite bhrātuḥ</i> 5.4.157
parvata	<i>trikakutparvate</i> 5.4.147
mitra/amitra	<i>suhṛd durhṛdau mitrāmitrayoḥ</i> 5.4.150
avasthā	<i>kakudasyāvasthāyām lopah</i> 5.4.146
vayaḥ	<i>vayasi dantasya datṛ</i> 5.4.141
lubdhayogaḥ	<i>dakṣiṇermā lubdhayoge</i> 5.4.126
No extra information needed for implementation	
<i>uttaramṛgapūrvācca sakthnaḥ</i> 5.4.98	
<i>anaśca</i> 5.4.108	
<i>napuṃsakādanyatarasyām</i> 5.4.109	
<i>nadīpaurṇamāsyāgrahāyaṇībhyaḥ</i> 5.4.110	
<i>jhayaḥ</i> 5.4.111	
<i>gireśca senakasya</i> 5.4.112	

<i>ahassarvaikadeśa saṅkhyātapuṇyācca rātreḥ 5.4.87</i>
<i>rājāhaḥ sakhibhyaṣṭac 5.4.91</i>
<i>grāmakauṭābhyāṃ ca takṣṇaḥ 5.4.95</i>
<i>ateḥ śunaḥ 5.4.96</i>
<i>nāvodvigoh 5.4.99</i>
<i>khāryāḥ prācām 5.4.101</i>
<i>dvitribhyāmañjaleḥ 5.4.102</i>
<i>kumahadbhyāmanyatarasyām 5.4.105</i>
<i>nadyrtaśca 5.4.153</i>
<i>aṅgulerdāruṇi 5.4.114</i>
<i>dvitribhyāṃ śa mūrdhnaḥ 5.4.115</i>
<i>antarbahirbhyāṃ ca lomnaḥ 5.4.117</i>
<i>upasargācca 5.4.119</i>
<i>suprātasuśvasudiva 5.4.120</i>
<i>nañ duḥsubhyo halisakthyoranyatarasyām 5.4.121</i>
<i>nityamasicprajāmedhasoḥ 5.4.122</i>
<i>dharmādanickevalāt 5.4.124</i>
<i>jambhāsuharitatṛṇasomebhyaḥ 5.4.125</i>
<i>prasambhyāṃ jānunorjñuḥ 5.4.129</i>
<i>ūrdhvādvibhāṣā 5.4.130</i>
<i>dhanuṣaśca 5.4.131</i>
<i>jāyāyā niṇ 5.4.134</i>
<i>gandhasyedutpūtisusurabhibhyaḥ 5.4.135</i>
<i>vibhāṣā śyāvārokābhyām 5.4.144</i>

<i>agrānta-śuddha-śubhra-vr̥ṣa-varāhebhyaśca 5.4.145</i>
<i>udvidbhyāṃ kākudasya 5.4.148</i>
<i>pūrṇādvibhāṣā 5.4.149</i>
<i>inaḥ striyām 5.4.152</i>
<i>tyasaśca 5.4.156</i>
<i>niṣpravāṇiśca 5.4.160</i>
<i>dvandvāccudaṣahāntātsamāhāre 5.4.106</i>
<i>ṛkpūrabdhūḥ pathāmānakṣe 5.4.74</i>
<i>acpratyanvavapūrvātsāmalomnaḥ 5.4.75</i>
<i>acaturavicitura 5.4.77</i>
<i>brahmahastibhyāṃ varcasah 5.4.78</i>
<i>avasamandhebhyaśtamasaḥ 5.4.79</i>
<i>anvavataptādrahasah 5.4.81</i>
<i>praterurasaḥ saptamīsthāt 5.4.82</i>
<i>dvistāvātristāvā vedih 5.4.84</i>
<i>gorataddhitaluki 5.4.92</i>
<i>alpākhyāyām 5.4.136</i>
<i>upasargādadhvanaḥ 5.4.85</i>
<i>nañastatpuruṣāt 5.4.71</i>
<i>patho vibhāṣā 5.4.72</i>
<i>śeṣādvibhāṣā 5.4.154</i>

pūrvanipātaḥ	
Input needed for implementation	sutras
gaṇapāṭhaḥ	<i>rājadantādiṣu param</i> 2.2.31
	<i>vā”hitāgnyādiṣu</i> 2.2.37
	<i>kaḍārāḥ karmadhāraye</i> 2.2.38
	<i>upamitaṁ vyāghrādibhiḥ</i> <i>sāmānyāprayoge</i> 2.1.56
Semantic category	
viśeṣaṇavācī	<i>saptamīviśeṣaṇe bahuvrīhau</i> 2.2.35
upamānavācakāḥ	<i>upamānāni</i> <i>sāmānyavacanaiḥ</i> 2.1.55
No extra information needed for implementation	
<i>upasarjanaṁ pūrvam</i> 2.2.30	
<i>dvandve ghi</i> 2.2.32	
<i>ajādyadantam</i> 2.2.33	
<i>alpāctaram</i> 2.2.34	
<i>niṣṭhā</i> 2.2.36	
<i>pūrvakālaika-sarva-jarat-purāṇa-nava-kevalāḥ samānādhikaraṇena</i> 2.1.49	
<i>pūrvāpara-prathama-carama-jaghanyasamāna-madhya-madhyamottama-vīrāśca</i> 2.1.58	
<i>yuvā-khalati-palita-valina-jaratībhiḥ</i> 2.1.67	

aluk samāsaḥ	
Input needed for implementation	sutras
ākroṣa	ṣaṣṭhyāḥ ākroṣe 6.3.21
Semantic category	
saṃñhāvācī	haladantātsaptamyāḥ sañjñayām 6.3.9
	manasaḥ saṃñāyām 6.3.4
	vaiyākaraṇākhyāyām caturthyāḥ 6.3.7
	parasya ca 6.3.8
kāravācī	kāranāmnī ca prācāṃ halādaḥ 6.3.10
svāṅgavācī	amūrdhamastakāt svāṅgādakāme 6.3.12
pūraṇavācī	ātmanaśca 6.3.6
kālavācī	śayavāsavāsiṣvakālāt 6.3.18
vidyāvācī/yonivācī	ṛto vidyā-yoni- sambandhebhyaḥ 6.3.23

No extra input needed for implentation
<i>pañcamyāḥ stokādibhyaḥ 6.3.2</i>
<i>ojaḥ saho'mbhas-tamasas-trtīyāyāḥ 6.3.3</i>
<i>ājñāyini ca 6.3.5</i>
<i>madhyād gurau 6.3.11</i>
<i>bandhe ca vibhāṣā 6.3.13</i>
<i>prāvṛtcharatkāladivām je 6.3.15</i>
<i>vibhāṣa vṛkshaśaravarāt 6.3.16</i>
<i>ghakālataneṣu kālanāmnaḥ 6.3.17</i>
<i>sthe ca bhāṣāyām 6.3.20</i>
<i>putre'nyatarasyām 6.3.22</i>
<i>nensiddhabadhnātiṣu ca 6.3.19</i>
<i>vibhāṣā svasṛpatyoḥ 6.3.24</i>
<i>tatpuruṣe kṛti bahulam 6.3.14</i>

pūrvapadakārya	
Input needed for implementation	sutras
vṛddhi	<i>idvṛddhau</i> 6.3.28
	<i>vṛddhinimittasya ca taddhi-</i> <i>tasyāraktavikāre</i> 6.3.39
Semantic category	
devatāvācī	<i>ānaṅṛto dvandve</i> 6.3.25
	<i>devatādvandve ca</i> 6.3.26
	<i>īdagneḥ somavaruṇayoḥ</i> 6.3.27
	<i>divo dyāvā</i> 6.3.29
	<i>uṣāsoṣasaḥ</i> 6.3.31
sañjñā/pūraṇavācī	<i>sañjñāpūraṇayośca</i> 6.3.38
svāṅgavācī	<i>svāṅgāccetaḥ</i> 6.3.40
jātivācī	<i>jātesca</i> 6.3.41
kālavācī	<i>avyayībhāve cākāle</i> 6.3.81
pūraṇa/priyādi	<i>striyāḥ puṃvat</i> <i>bhāṣītapuṃskādanūṇ</i> <i>samānādhikaraṇe</i> <i>striyāmappūraṇīpriyādiṣu</i> 6.3.34

No extra information needed for implementation
<i>vopasarjanasya 6.3.82</i>
<i>vibhaṣā catvāriṃśatprabhṛtau sarveśām 6.3.49</i>
<i>trestrayaḥ 6.3.48</i>
<i>dvyasṭanaḥ saṃkhyāyāmabahuvrīthyaśītyoḥ 6.3.47</i>
<i>puṇvatkarmadhāraya-jātīya-deśīyeṣu 6.3.42</i>
<i>ānmahataḥ samānādhikaraṇajātīyayoḥ 6.3.46</i>
<i>divasaśca pṛthivyām 6.3.30</i>
<i>mātarapitaravudīcām 6.3.32</i>
<i>tasilādiṣvākṛtvasucaḥ 6.3.35</i>
<i>kyaṇmāninośca 6.3.36</i>
<i>na kopadhāyāḥ 6.3.37</i>

uttarapadākārya	
Input needed for implementation	sutras
gaṇapāṭhaḥ	kṣubhnādiṣu ca 8.4.39
Semantic category	
sañjñāvācī	pūrvapadātsañjñāyāmagāḥ 8.4.3
saṅkhyā	na saṅkhyādeḥ samāhāre 5.4.89
No extra information needed for implementation	
ahno'hna etebhyaḥ 5.4.88	
uttamaikābhyām ca 5.4.90	
upasargācca 5.4.119	
ahno'dantāt 8.4.7	
upasargād bahulam 8.4.28	
puraḡāmiśrakā-sidhrakā-sārikā-kotarāgrebhyaḥ 8.4.4	
pranirantaḥśarekṣu-plakṣāmra-kārṣya-khadira-pīyūkṣābhyo'sañjñāyāmapi 8.4.6	
jyotirāyuṣaḥ stomaḥ 8.3.83	

samāsāśraya prakaraṇam	
Input needed for implementation	sutras
gaṇapāṭhaḥ	vanagiryoh sañjñāyām koṭarakiṃśulukādīnām 6.3.117
	matau bahvaco' najirādīnām 6.3.119
	śarādīnām ca 6.3.120
Semantic category	
ṛk vācakāḥ	ṛcaḥ śe 6.3.55
sañjñāvācī	udakasyodaḥ sañjñāyām 6.3.57
	nyāpoḥ sañjñāchandasorbahulam 6.3.63
	sahasya saḥ sañjñāyām 6.3.78
	aṣṭanaḥ sañjñāyām 6.3.125
	nare sañjñāyām 6.3.129
pūrayitavyavācī	ekahalādau pūrayitavye'nyatarasyām 6.3.59
granthavācī	granthāntādhike ca 6.3.79

anumeyah	<i>dvitīye cānupākhye</i> 6.3.80
caraṇavācī	<i>caraṇe brahmacāriṇi</i> 6.3.86
jātivācī	<i>trṇe ca jātau</i> 6.3.103
	<i>kustumburuṇi jātiḥ</i> 6.1.143
rṣivācakāḥ	<i>mitre cārṣau</i> 6.3.130
Morph analysis is needed	
<i>gharūpa-kalpa-celaḍ-bruva-gotra-mata-hateṣu ṇyo'nekāco hrasvaḥ</i> 6.3.43	
<i>ugitaśca</i> 6.3.45	
<i>hṛdayasya hṛllekhayadaṇlāseṣu</i> 6.3.50	
<i>vā śokaṣyañrogeṣu</i> 6.3.51	
<i>iko hrasvo'ṇyo gālavasya</i> 6.3.61	
<i>ṣyaṇaḥ samprasāraṇaṁ putrapatyostatpuruṣe</i> 6.3.13	
<i>rātreḥ kṛti vibhāṣā</i> 6.3.72	
<i>tīrthe ye</i> 6.3.87	
<i>vibhāṣodare</i> 6.3.88	
<i>dṛgadṛśavatuṣu</i> 6.3.89	

No extra information needed for implementation
<i>nadyāḥ śeṣasyānyatarasyām 6.3.44</i>
<i>himakāṣihatiṣu ca 6.3.54</i>
<i>vā ghoṣamiśraśabdeṣu 6.3.56</i>
<i>peṣaṃvāsavāhanadhiṣu ca 6.3.58</i>
<i>manthaudana-saktu-bindu-vajrabhāra-hāra-vīvadhagāheṣu ca 6.3.60</i>
<i>pādasya padājyātigopahateṣu 6.3.52</i>
<i>iṣṭakeṣīkamālānām citatūlabhāriṣu 6.3.65</i>
<i>kāre satyāgadasya 6.3.70</i>
<i>samānasya cchandasyamūrdhaprabhṛtyudarkeṣu 6.3.84</i>
<i>jyotirjanapada 6.3.85</i>
<i>idaṅkimorīśkī 6.3.90</i>
<i>aṣaṣṭhyatṛtīya 6.3.99</i>
<i>koḥ kattatpuruṣe'ci 6.3.101</i>
<i>rathavadayośca 6.3.102</i>
<i>kā pathyakṣayoḥ 6.3.104</i>
<i>īṣadarthe 6.3.105</i>
<i>vibhāṣā puruṣe 6.3.106</i>
<i>kavaṃ coṣṇe 6.3.107</i>
<i>karṇe lakṣaṇasyā 6.3.115</i>
<i>nahivṛtivrṣiviyadhirucisahitaniṣu kvau 6.3.116</i>
<i>vale 6.3.118</i>
<i>iko vahe'pīloḥ 6.3.121</i>
<i>upasargasya ghaṇyamanuṣye bahulam 6.3.122</i>
<i>ikaḥ kāśe 6.3.123</i>
<i>citeḥ kapi 6.3.127</i>

<i>aparasparāḥ kriyāsātatyē 6.1.144</i>
<i>goṣpadaṃ sevitāsevitapramāṇeṣu 6.1.145</i>
<i>āspadaṃ pratiṣṭhāyām 6.1.146</i>
<i>āścaryamanitye 6.1.147</i>
<i>varcaske'vaskaraḥ 6.1.148</i>
<i>apaskaro rathāṅgam 6.1.149</i>
<i>viṣkīraḥ śakunau vā 6.1.150</i>
<i>pratiṣkaśaśca kaśeḥ 6.1.152</i>
<i>praskaṇvahariścandrāvṛṣī 6.1.153</i>
<i>maskaramaskariṇau veṇuparivrājakayoḥ 6.1.154</i>
<i>kāstīrājastunde nagare 6.1.155</i>
<i>kāraskaro vṛkṣaḥ 6.1.156</i>

liṅganirdhāraṇakārya	
Input needed for implementation	sutras
gaṇapāṭhaḥ	<i>ardharcāḥ puṃsi ca</i> 2.4.31
	<i>na dadhipayaādīni</i> 2.4.14
	<i>gavāśvaprabhṛtīni ca</i> 2.4.11
Semantic category	
jātivācī	<i>jātyākhyāyām-ekasmin- bahuvacanam-anyatarasyām</i> 1.2.48
niravasitāḥ	<i>śūdrānāmaniravasitānām</i> 2.4.10
vṛkṣa/mṛga/tṛṇa/dhānya	<i>vibhāṣā vṛkṣa-mṛga-tṛṇa-dhānya- vyañjana-paśuśakunyaśvavadava- pūrvāparādharrattarāṇām</i> 2.4.12
adhikaraṇavāci	<i>vipratīṣiddham cānadhikaraṇavāci</i> 2.4.13
	<i>adhikaraṇaitāvattve ca</i> 2.4.15
sañjñāyā vācī	<i>sañjñāyām kanthośīnareṣu</i> 2.4.20
prāṇi/tūrya/senāṅga	<i>dvandvaśca prāṇitūryasenāṅgānām</i> 2.4.2
caraṇa	<i>anuvāde caraṇanām</i> 2.4.3
adhvaryukratu	<i>adhvaryukraturanapuṃsakam</i> 2.4.4
viprakṛṣṭākhyāyā	<i>adhyayayanato 'viprakṛṣṭākhyāyām</i> 2.4.5
jāti/prāṇivācī	<i>jātiraprāṇinām</i> 2.4.6
nadī/deśa/grāmāḥ	<i>viśiṣṭaliṅgonadīdeśo 'grāmāḥ</i> 2.4.7

kṣudrajantavaḥ	kṣudrajantavaḥ 2.4.8
nakṣatra vācākāḥ	phalgunīproṣṭhapadānāñca nakṣatre 1.2.60
No extra information needed for implementation	
	paravallīṅgaṃ dvandvatatpuruṣayoḥ 2.4.26
	pūrvavadaśvavaḍavau 2.4.27
	rātrāhnāhāḥ puṃsi 2.4.29
	apathaṃ napuṃsakam 2.4.30
	asmado dvayośca 1.2.49
	sa napuṃsakam 2.5.17
	upajñopakramaṃ tadādyācikyāsāyām 2.4.11
	sabhā rājā'manuṣyapūrvā 2.4.23
	vibhāṣā samīpe 2.4.1
	aśālā ca 2.4.24
	vibhāṣā senāsurācchāyāśālāniśānām 2.4.25
	chāyā bāhulye 2.4.22
	yeṣāṃ ca virodhaḥ śāśvatikaḥ 2.4.9

vacananirdhāraṇakārya	
Input needed for implementation	sutras
gaṇapāṭhaḥ	na dadhipayaādīni 2.4.14
Semantic category	
jāti vācī	jātyākhyāyām-ekasmin- bahuvacanam-anyatarasyām 1.2.48
adhikaraṇavāci	vipraṭiṣiddham cānadhikaraṇavāci 2.4.13
prāṇi tūryasenāṅgānāḥ	dvandvaśca prāṇitūryasenāṅgānām 2.4.2
caraṇa	anuvāde caraṇanām 2.4.3
adhvaryukratu	adhvaryukraturanapuṃsakam 2.4.4
viprakṛṣṭākhyāyā	adhyayayanato'viprakṛṣṭākhyāyām 2.4.5
jāti evam prāṇi vācī	jātiraprāṇinām 2.4.6
nadī deśa evam grāmāḥ	viśiṣṭaliṅgonadīdeśo'grāmāḥ 2.4.7
nakṣatra vācākāḥ	phalgunīproṣṭhapadānāñca nakṣatre 1.2.60
Pragmatics	
niravasitāḥ	sūdrānāmaniravasitānām 2.4.10

gatikārya	
Input needed for implementation	sutras
Semantics	
anukaraṇavācī	<i>anukaraṇaṃ cānitiparam 1.4.61</i>
ādara/anādara	<i>ādarānādarayoḥ sadasatī 1.4.62</i>
bhūṣaṇe	<i>bhūṣaṇe'lam 1.4.63</i>
aparigraha	<i>antaraparigrahe 1.4.64</i>
śraddhā/pratīghāta	<i>kaṇemanasi śraddhāpratīghāte 1.4.65</i>
aupamya	<i>jīvikopaniṣadāvaupamye 1.4.79</i>
upayamana	<i>nityaṃ haste pāṇāvupayamane 1.4.77</i>
bandhana	<i>prādhvaṃ bandhane 1.4.78</i>
anupadeśa	<i>ado'nupadeśe 1.4.69</i>
antardhi	<i>tiro'ntardhau 1.4.70</i>
No extra information needed for implementation	
<i>purovyayam 1.4.66</i>	
<i>astaṃ ca 1.4.67</i>	
<i>accha gatyarthavadeṣu 1.4.68</i>	
<i>vibhāṣā kṛñi 1.4.72</i>	
<i>upāje'nvāje 1.4.73</i>	
<i>anatyādhāna urasimanasī 1.4.75</i>	
<i>madhye pade nivecane ca 1.4.76</i>	

Chapter 5

Semantic and Pragmatic Information Needed for Compound Generation

In the previous chapter we noted down various kind of extra linguistic information needed for the implementation of sūtras. This extra linguistic information falls under two different categories viz semantic and pragamatic. The semantic type of information is the one which can be associated with the lexicon and is context independent, while the pragmatic information is context dependent. To give an example, “a component in the sense of distance (dūrārtha)” is a semantic condition, whereas “referent of a component is respectable” is a pragmatic condition because the referent depends on the context.

In order to provide the necessary semantic information with lexicon, we noticed that Pāṇini uses several semantic features. Those are

nadīvācī, deśavācī, grā mavācī, vidyāvācī, yonivācī, devatāvācī, vyañjanavācī, bhakṣyavācī, parvatavācī, ṛṣivācī, nakṣatravācī, saṅkhyāvācī, saṅkhyeyavācī, pūraṇavācī, catuṣpādvācī, varṇavācī, kṣudrajanu, vṛkṣavācī, tṛṇavācī, dhānyavācī, ṛṇavācī, janapadavācī, granthavācī, jāti, guṇavācī, dūrārthavācī, parimāṇavācī, dikvācī, kālavācī.

These features fall under the following sub-categories

- (a) Meaning
- (b) Synonym / antonym
- (c) Part-whole relation
- (d) Modifier-modified relation
- (e) Genre it belongs to

There have been several efforts to classify the objects in the real world based on the metaphysical properties of these objects. This classification is known as ontological classification. Some of the important efforts in this direction in the West are Roger's thesaurus, CYC¹ and SUMO².

In India the importance of such a classification had been recognized long-back by the Vaiśeṣikas. This classification is only a top level classification classifying the objects into seven classes viz dravya, guṇa, karma, sāmānya viśeṣa, samavāya and abhāva and each of these are further subclassified. However this classification needs to be further extended so as to classify all the real objects showing the hierarchical classification. But in order to extend the Vaiśeṣika classification, we need to understand the concept of Genre (jāti).

¹<http://www.cyc.com/>

²<http://www.adampease.org/OP/>

The sūtra ‘samāna prasavātmikā jātiḥ N2.2.71’ about jāti says that Universal is that which produces the same conception. Cowhood is an universal which resides in all cows. Assuming that if a person sees a cow somewhere then he acquires a general form of cows. This general form enables him to recognise individual cows, anywhere in the world. Any general form such as an X-ness can’t be considered to be a jāti. Because this will lead to multiple inheritance.

Udayana, the great Naiyāyika, has discussed about the six impediments (jātibādhakas) in Nyāyasiddhantamuktāvali to decide whether a given general property is an universal or not.

The six impediments are described in Muktāvali as

“vyakterabhedastulyatvaṃ saṅkaro‘thā‘navasthitiḥ
rūpahānirasambandho jātibādhakasamgrahaḥ”

Vyakteḥ abhedaḥ (Oneness of the individual), tulyatvam (Identity), saṅkaraḥ (cross-division), anavasthitiḥ (Infinite Regression), rūpahāniḥ (Violation of Nature) and asambandhaḥ (Non relation) are the six impediments (jātibādhakāḥ).

Vyakteḥ abhedaḥ (Oneness of the individual) : When there is a single entity of that kind, then the property of resident in only that object can’t be an Universal. For example, in Nyāya-Vaiśeṣika system ākāśa is only one hence ākāśatva cannot be a jāti.

Tulyatvam (Identity) : If two general properties occur in the same individuals, then these two properties are equivalent and hence only one among

these two will be considered to be a jāti. For example, ghaṭa and kalaśa refer to the same set of objects. Hence only one of them will be a jāti.

Saṅkara (Cross-division) : Physicalness (bhūtatva) and mortality (mūrtatva) are not universals because they involve cross-division. Bhūtatva is the universal property that exists in five elements. The five elements are pṛthvī (earth), jala (water), tejas (fire) and vāyu (air) and ākāśa. The first four also have mūrtatva (mortality). Apart from these four elements, the manas (mind) is also mūrta and hence it has mūrtatva. Now the question is whether the general properties mūrtatva and bhūtatva both represent a genre / class character (jāti) or not? Udayana puts a constraint that among the genre there should not be a saṅkara. In other words a padārtha should not have two generic properties residing in them. The four elements pṛthvī (earth), jala (water), tejas (fire) and vāyu (air) have two universals (bhūtatva and mūrtatva) in them and this leads to saṅkara, and hence both mūrtatva and bhūtatva can't be considered to be jātis simultaneously. This condition is an essential characterisation of single inheritance in the Western terminology.

Anavasthitiḥ (Infinite regress): A universal shouldn't contain another universal, as it may lead to anavasthitiḥ (Infinite regress).

Rūpahāniḥ (Violation of Nature) : Differentials, cannot have an universal, like differential-ness (viśeṣatva). If we consider viśeṣatva as an identity or universal, then mutual difference would be there.

Asambandhaḥ (Non relation): Inherence cannot have a universal, like inherence-ness (samavāyatva). Inherence is a relation which holds universal in a particular. Assume that if inherence-ness is regarded as a universal, then inhering in inherence (samavāya) will lead to infinite regress.

So as an abstract property of an object can be a *jāti* provided it does not fall under any of these impediments. Any generic property that does not satisfy any of the above *jātibādhakās* will then be termed as an *upādhi*.

The Nyāya-Vaiśeṣika school of philosophy in Indian tradition has dealt with the ontological classification of the real objects which is accepted to all the branches in Indian knowledge system. All the words in the Amarakośa were classified following and extending this scheme further by Nair (2011). To this classification, she also added various other properties, called as *upādhis*, a word may have other than its ontological status. For example, to be an eatable, or to be drinkable are the special properties which can not be classified under the ontology without violating the essential property of a tree viz. the single inheritance. We examined all the semantic conditions found in all the sūtras related to compound generation. Some ontological categories (*jātis* as well as *upādhis*) which are used by Pāṇini were missing in the tables developed by Nair, since these were basically prepared using the lexicon from Amarakośa. We added these missing categories to the table of ontological classification (Figure 5.1) and the missing properties to the table of *upādhis* (Figure 5.2). The lexicon enriched with the ontological and special properties belonging to the *upādhi* chart is used for the compound generation.

5.1 List of jātis needed for Compound Generation

nadīvācī

nadībhiśca 2.1.19

viśiṣṭaliṅgonadīdeśo'grāmāḥ 2.4.7

deśavācī

viśiṣṭaliṅgonadīdeśo'grāmāḥ 2.4.7

grāmāvācīḥ

viśiṣṭaliṅgonadīdeśo'grāmāḥ 2.4.7

devatāvācī

ānanṛto dvandve 6.3.25

devatādvandve ca 6.3.26

īdagneḥ somavarūṇayoḥ 6.3.27

dīvo dyāvā 6.3.29

uṣāsoṣasaḥ 6.3.31

parvatavācī

trikakutparvate 5.4.147

ṛṣi

mitre cārṣau 6.3.130

nakṣatravācī

phalgunīproṣṭhapadānāñca nakṣatre 1.2.60

saṅkhyāvācī

akṣaśalākāsaṅkhyāḥ pariṇā 2.1.10

saṅkhyā vaṁśyena 2.1.18

saṅkhyāpūrvō dviguḥ 2.1.51

saṅkhyāvyayāsanna-dūrādhika-saṅkhyāḥ saṅkhyeye 2.2.25

tatpuruṣasyāṅguleḥ saṅkhyāvyayādeḥ 5.4.86

saṅkhyā supūrvasya 5.4.140

na saṅkhyādeḥ samāhāre 5.4.89

diksaṅkhye sañjñāyām 2.1.50

saṅkhyeyavācī

bahuvrīhau saṅkhyeye ḍajabahugaṇāt 5.4.73

pūraṇa

ātmanaśca 6.3.6

sañjñāpūraṇyośca 6.3.38

ātmanaśca 6.3.6

pūraṇa-guṇa-suhitārtha-sadavyaya-tavya-samānādhikaraṇena 2.2.11

striyāḥ puṁvat bhāṣitapuṁskādanūñ samānādhikaraṇe

striyāmappūraṇīpriyādiṣu 6.3.34

catuṣpādvācī

catuṣpādo garbhīṇyā 2.1.71

varṇavācī

varṇo varṇena 2.1.69

kṣudrajantu

kṣudrajantavaḥ 2.4.8

vṛkṣavācī

vibhāṣā *vṛkṣa-mṛga-tṛṇa-dhānya-vyañjana-paśuśakunyaśvavadava-*
pūrvāparādharttarāṇām 2.4.12

tṛṇavācī

vibhāṣā *vṛkṣa-mṛga-tṛṇa-dhānya-vyañjana-paśuśakunyaśvavadava-*
pūrvāparādharttarāṇām 2.4.12

janapada

brahmaṇo janapadākhyāyām 5.4.104

grantha

granthāntādhike ca 6.3.79

kāla

ktenāhorātrāvayavāḥ 2.1.45

śayavāsavāsiṣvakālāt 6.3.18

avyayībhāve cākāle 6.3.81

svāṅga

amūrdhamastakāt svāṅgādakāme 6.3.12

svāṅgāccetaḥ 6.3.40

nāḍītantryoḥ svāṅge 5.4.159

bahuvrīhau sakthyakṣṇoḥ svāṅgātṣac 5.4.113

jāti

katarakatamau jātiparipraśne 2.1.63

poṭāyuvati 2.1.65

kṛtyatulyākhyā ajātyā 2.1.68

jātiraprāṇinām 2.4.6

jātyākhyāyām-ekasmin-bahuvacanam-anyatarasyām 1.2.48

ano'smāyassarasām jātisamjñayoḥ 5.4.94

jātesca 6.3.41

trṇe ca jātau 6.3.103

kustumburuṇi jātiḥ 6.1.143

jātyākhyāyām-ekasmin-bahuvacanam-anyatarasyām 1.2.48

jātiraprāṇinām 2.4.6

guṇavācī

trṭīyā tatkr̥tārthena guṇavacanena 2.1.29

dūrārtha

stokāntikadūrārthakṛcchrāṇi ktena 2.1.38

parimāṇa

kālāḥ parimāṇinā 2.2.5

dik

dinnāmānyantarāle 2.2.26

prāṇi

dvandvaśca prāṇitūryasenāṅgānām 2.4.2

upamānādapraṇiṣu 5.4.97

caraṇa

anuvāde caraṇānām 2.4.3

āyāmavācī

anugavamāyāme 5.4.83

ṛk vācakāḥ

ṛcaḥ śe 6.3.55

pūrayitavya vācī

ekahalāḍau pūrayitavye'nyatarasyām 6.3.59

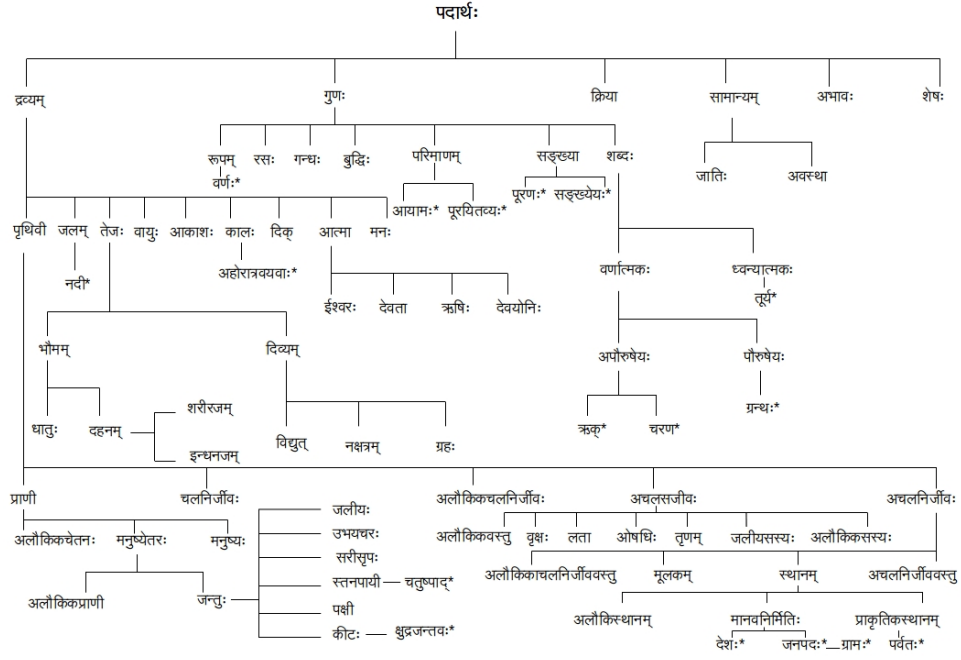


Figure 5.1: Ontological classification

5.2 List of Upādhis needed for Compound Generation

vidyāvācī

ṛto vidyāyonisambandhebhyaḥ 6.3.23

yonivācī

ṛto vidyāyonisambandhebhyaḥ 6.3.23

dhānyavācī

vibhāṣā vṛkṣa-mṛga-trṇa-dhānya-vyañjana-paśuśakunyaśvavadava-

pūrvāparādharttarāṇām 2.4.12

vyañjanavācī

annena vyañjanam 2.1.33

vibhāṣā vṛkṣa-mṛga-tṛṇa-dhānya-vyañjana-paśuśakunyaśvavadava-
pūrvāparādharttarāṇām 2.4.12

bhakṣyavācī

bhakṣyeṇa miśṛikaraṇam 2.1.34

vayas

vayasi dantasya datṛ 5.4.141

ṛṇa

kṛtyairṛṇe 2.1.43

niravasitāḥ

śūdrānāmaniravasitānām 2.4.10

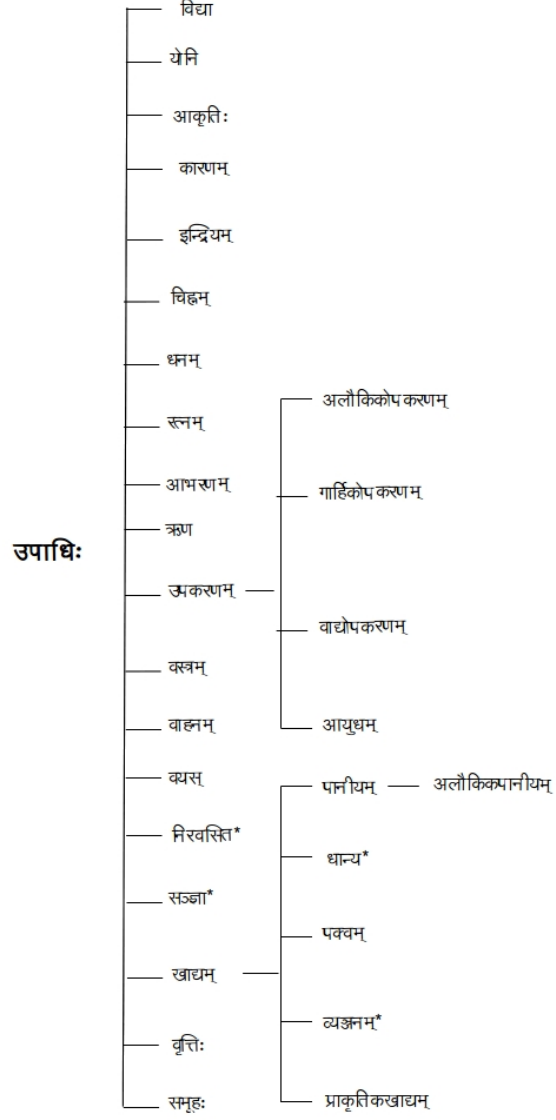


Figure 5.2: Other properties

5.3 Pragmatics

There are several cases we encountered in the derivation process which required context dependent information, which can't be available through a lexicon. For example, if the resulting compound denotes a name *saṃñā* or not. This is purely context dependent and we need to get this information from the user who desires to make a compound. Various categorial information that is context dependent needed for compound generation is listed below, giving the list of relevant sūtras which need this information.

sañjñāvācī

haladantātsaptamyāḥ sañjñāyām 6.3.9

manasaḥ saṃñāyām 6.3.4

vaiyākaraṇākhyāyāṃ caturthyāḥ 6.3.7

parasya ca 6.3.8

sañjñāyām 2.1.44

anyapadārthe ca sañjñāyām 2.1.20

aññāsikāyāḥ sañjñāyāṃ nasaṃ cāsthūlāt 5.4.118

vā sañjñāyām 5.4.133

na sañjñāyām 5.4.155

agrākhyāyāmurasāḥ 5.4.93

striyāṃ sañjñāyām 5.4.143

pūrvapadātsañjñāyāmagāḥ 8.4.3

sañjñāyāṃ kanthośīnareṣu 2.4.20

udakasyodaḥ sañjñāyām 6.3.57

nyāpoḥ sañjñāchandasorbahulam 6.3.63

sahasya saḥ sañjñāyām 6.3.78

aṣṭanaḥ sañjñāyām 6.3.125

nare sañjñāyām 6.3.129

kāravācī

kāranāmnī ca prācāṃ halāḍau 6.3.10

praśamsāvacana

praśamsāvacanaiśca 2.1.66

adhvaryukratuḥ

adhvaryukraturanapuṃsakam 2.4.4

sādrśye

yathā'sādrśye 2.1.7

kṣepe

khaṭvā kṣepe 2.1.25

kṣepe 2.1.47

kiṃ kṣepe 2.1.64

kimāḥ kṣepe 5.4.70

atyantasamyoge

atyantasamyoge ca 2.1.28

adhikārthavacana

kṛtyairadhikārthavacane 2.1.32

upamānavācakāh

upamānāni sāmānyavacanaiḥ 2.1.55

upamānācca 5.4.137

pūjā

ktena ca pūjāyām 2.2.12

na pūjanāt 5.4.69

pūjyamāna

sanmahatparamottamotkrṣṭāḥ pūjyamānaiḥ 2.1.61

vṛndarakanāgakuñjaraiḥ pūjyamānam 2.1.62

adhikaraṇavācī

adhikaraṇavācinā ca 2.2.13

vipratīṣiddham cānadhikaraṇavāci 2.4.13

adhikaraṇaitāvattve ca 2.4.15

ekadeśi

pūrva-parādhharottaram-ekadeśinaikādhikaraṇe 2.2.1

karmavyatihārah

ickarmavyatihāre 5.4.127

adarśanam

akṣṇo'darśanāt 5.4.76

vandanam

vandite bhrātuh 5.4.157

lubdhayogaḥ

dakṣiṇermā lubdhayoge 5.4.126

viśeṣaṇavācī

saptamīviśeṣaṇe bahuvrīhau 2.2.35

anumeyaḥ

dvitīye cānupākhye 6.3.80

viprakṛṣṭākhyā

adhyayayanato 'viprakṛṣṭākhyāyām 2.4.5

avasthā

kakudasyāvasthāyām lopah 5.4.146

mitra

suhṛd durhṛdau mitrāmitrayoḥ 5.4.150

amitra

suhṛd durhṛdau mitrāmitrayoḥ 5.4.150

Chapter 6

Implementation Details

In the second chapter we saw the steps involved in the compound word /stem generation. The process starts with an *alaukikavigraha* (analytical paraphrase) and undergoes a series of transformations, resulting into a compound. Each transformation corresponds to a well defined operation in the process. Now we describe each module. We give below the output of each module the range of *sūtras* that correspond to this operation, the linguistic resources needed for their implementation. The output of the previous module is the input for the successive module.

6.1 Description of modules and their implementation

A) **Module:** Designating a *saṃjñā alaukikavigraha*

Input: A string in either of the following forms

stem '+' sup stem '+' sup

avyaya stem '+' sup
stem '+' sup avyaya

Output: Designation of the sañjñā alaukikavigraha.

At this stage the input string is tokenized into stem1, stem2, sup1, sup2, avyaya etc as the case may be. Every stem acquires the semantic features such as gender, meaning, gaṇa, jāti and upādhi.

Resources: A structured lexicon with the following information for each stem/prātipadikam: liṅgam, meaning, gaṇa, jāti, upādhi, vacanam

Pāṇinian sūtra: saha supa (A2.1.4)

This step may well be implemented as a context free grammar rewrite rule in say, Yacc, with a lexer to tokenize the input. The lexer returns a feature structure from the lexicon along with each token identified. The grammar is written as

alaukikavigrahaḥ : stem '+' sup stem '+' sup
 |avyaya stem '+' sup
 |stem '+' sup avyaya
 ;

B) **Module:** Designating a sañjñā **samāsa** for the analytical paraphrase

Output: Designation of the samāsa sañjñā for the alaukikavigraha

Pāṇinian sūtra: prākkadārātsamāsaḥ (A2.1.3)

This also can be implemented simply by a context free rewriting rule that assigns a new designation as

samāsa : alaukikavigrahaḥ
 ;

C) Module: Introduction of the **samāsanta suffixes**

The introduction of samāsanta suffix is under specific conditions. As we saw in the previous chapter, these conditions are either simple conditions on the input string comprising of alaukikavighraha or may need some morphological, semantic or pragmatic information. Further, among all the sūtras at the most one sūtra is applicable. So we decided to implement this module as a lexer which has a powerful regular expression matching which can take care of the conditions on the input string. Other conditions on morphology, semantics and pragmatics are implemented as an Ansi C code. At this stage a samāsanta suffix is added to the alaukika vighraha.

Output: alaukikavighraha samāsantasuffix.

Pāṇinian sūtra: The sūtras from A5.4.68 to A5.4.160 are related to this samāsanta module.

D) Module: Designating **prātipadikasañjñā** to the compound

Output: Designation of the prātipadikasañjñā for the compound.

Pāṇinian sūtra: kṛttaddhitasamāsāsca (A1.2.46)

This again is just assignment of a prātipadikasañjñā which can be implemented naturally as a rewrite rule as

prātipadikam : samāsa

;

E) Module: Designating **upasarjanasañjñā** to decide the word order

Output: index of the stem getting an upasarjana designation.

Pāṇinian sūtras: prathamānirdiṣṭam samāsa upasarjanam (A1.2.43), and all samāsavidhāyaka sūtras.

In this Module, the order of the components in the compounded form is decided. The relevant sūtras are ‘prathamānirdiṣṭam samāsa upasarjanam’ (A1.2.43) and ‘upasarjanam pūrvam’ (A2.2.30). By the first sūtra first the word that takes the position is assigned an upasarjanasaṃjñā and later it is moved to the first place. The upasarjanasaṃjñā is assigned to the word referred to by the nominative case ending term in the sūtra. In case of Bahuvrīhi and Dvandva, we cannot assign the upasarjanasaṃjñā. The governing sūtra in Bahuvrīhi is ‘anekamanyapadarthe’ A2.2.24. Here the word in nominative case is ‘anekam’ which refers to all the words together (undergoing the compound formation). Similarly in the case of Dvandva, the governing rule ‘cārthe dvandvaḥ’ A2.2.29. In this sūtra there is no such word which ends in nominative case as in other governing sūtras. Hence, there is no scope of upasarjanasaṃjñā in these two compound types. In such cases Pāṇini gives rules to decide the first component (pūrvanipata).

In case of tatpuruṣa compounds one sūtra deserves special treatment. The sūtra is ‘kugatiprādayaḥ’ A2.2.18. Here in this sūtra Pāṇini makes use of a special technical term ‘gati’. This sūtra says in an alaukikavigraha, the component with gatisaṃjñā gets an upasarjanasaṃjñā. Hence the algorithm for deciding the position of the component assuming first place is as follows.

1. If the samāsaprakāra is Tatpuruṣa and the samāsavidhāyaka sūtra is ‘kugatiprādayaḥ’ A2.2.18 then get the position of the component with gatisaṃjñā assign upasarjanasaṃjñā to the component with gatisaṃjñā
2. In all other cases of tatpuruṣa, and in the case of Avyayībhāva get the position of a component qualifying the upasarjanasaṃjñā.
3. In case of Bahuvrīhi
 - (1) If there is a term in 7th case, it gets the 1st position.

- (2) If there is viśeṣaṇa-viśeṣyabhāva, the component which is viśeṣaṇa gets the 1st position.

4. In case of Dvandva

- (1) If a word, which ends in 'ghi' it gets the 1st position.
- (2) If a word, which starts with a vowel and ends in 'a' then it gets the 1st position.
- (3) If a word which contains less number of vowels will get the 1st position.
- (4) The compounds which contains gaṇa like rājadantādi are treated as exceptions.

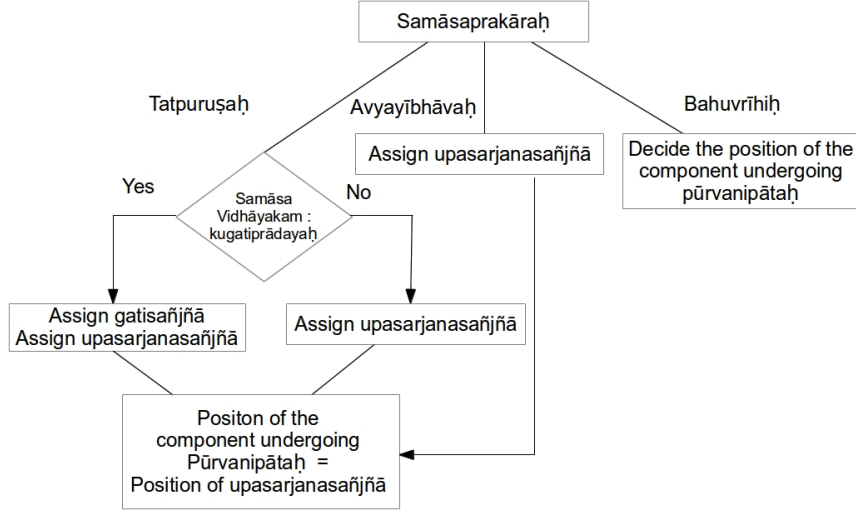


Figure 6.1: Deciding the position of the component

F) Module: Pūrvanipāta

Output: The alaukikavigraha in desired word order.

If an upasarjanasaṅjñā has been assigned, it gets the 1st position by ‘upasarjanam pūrvam’ (A2.2.30). In the absence of upasarjanasaṅjñā the word marked as pūrvanipāta assumes the 1st position.

G) Module: Aluk (Non-elision of the Vibhakti)

Output: The suffix of the 1st component is retained if certain conditions are satisfied.

Pāṇinian sūtra: The sūtras from A6.3.1 to A6.3.25 are related to this Aluk Module.

H) **Module: Subluk** (Elision of the vibhaktis)

Output: If either no sūtra related to aluk is applicable, or if aluk related sūtra is optionally applicable, then the deletion of the case suffixes takes place.

Pāṇinian sūtra: supo dhātuprātipadikayoḥ (A2.4.71)

This is achieved just by assigning a null value to the suffix.

I) **Module:** Transformation of the first component

In certain cases the first component undergoes certain transformations in the presence of the second component.

Pāṇinian sūtras: This module deals with the sūtras starting from the 1st and 3rd quarter of the 6th chapter

These are implemented using a lexer with pattern matching for the condition on the 2nd component and Ansi C functions for morphological, semantic or pragmatic conditions on the 2nd component.

J) **Module:** Transformation of the second component

In some cases the second constituent undergoes certain transformations in the presence of the first component

Pāṇinian sūtras: This module deals with some sūtras in the 8th chapter's 4th pada

of the Aṣṭādhyāyī. These sūtras belong to the tripādī section of Aṣṭādhyāyī. The rules in this section behave in a different way than those in the rest part of the Aṣṭādhyāyī. The main guiding sūtra for this section is 'pūrvatrāsiddham' (A8.2.1). This says that a sūtra in this section is invisible to all the sūtras following it. In other words, these sūtras are to be operated sequentially. Now a lexer allows application of only one sūtra. If any

other sūtra is applied on the output, we need to call the lexer again and this time we also have to remember the previous sūtra number. In the case of compound generation we observed that on a given string only one sūtra will be applicable and hence we could again implement these sūtras using lexer.

K) Module: Padakārya

In this module a special sūtra that transforms a pada is implemented using lex.

Pāṇinian sūtra: nalopaḥ prātipadikāntasya (A8.2.7)

L) Module: Sandhikārya

Here the two constituents are now joined following the sandhi operation.

We used the existing sandhi module for this compound generation.

6.2 Scope of Generator

We have implemented the modules upto sandhi. After this there are other operations such as deciding the gender (Liṅganirdhāraṇakārya), deciding the number (Saṅkhyā), Deciding the accent (Svara) which are not implemented. To implement the liṅganirdhāraṇakārya module we require a morphological analyser which can analyse the words with feminine ending suffixes and identify the feminine suffix. Since such an analyser is not in place, we decided to limit ourselves to the formation of the prātipadikam.

6.3 Some specific issues

1. Order of implementation of rules

While some of the modules were written as rewrite rules, most of the others are implemented using a lexer. Each module implemented with a lexer is thus a finite state automaton. When we decided to use the lexer such as flex (fast lexical analyser for GNU linux), we had to take into account the conflict resolution strategy used in flex. The conflict resolution in flex works as below.

- i) In case of more than one applicable rules, the one with largest matching string is tried.
- ii) In case of two rules with same length of matching expression, the earlier rule is tried.

Pāṇini typically states the general rule (utsarga) first and then he states the exception (apavāda). In case the exceptional rule involves some extra conditions then we can follow the given sequence of the rules. But in cases where the exception is stated for specific words, then we cannot write the exception rule after the general rule, in lex. Because in that case, since the length of the matching string is same, the earlier rule, i.e, the general rule will be tried and not the exceptional one. In such cases we have inverted the order of rules. In all other cases the order of the order in Aṣṭādhyāyī is followed. Here is an example:

The sūtra ‘ateḥ śunaḥ 5.4.96’ says that if the first word is ati and second word is śvan then an additional suffix ‘tac’ is added to

the entire string. The second sūtra ‘na pūjanāt 5.4.69’ says that if the first word is either su or ati and if it is in the sense of pūjā and is followed by any word then the suffix ‘tac’ is not added. If we write these rules in Lex as per Aṣṭādhyāyī then 5.4.69 takes place first and it does not allow the other rule 5.4.96 for the ‘tac’ operation by 5.4.96.

6.4 Evaluation

There is no such book apart from Pāṇinīyavyākaraṇodāharaṇakośa which deals with the entire process of compounds. So this is the only authenticated and manually tested data. Hence this is a gold data for our purpose. We have tested around 500 examples from this book and compared the output generated by our Sanskrit Compound Generator with it. The final results match. However, there are differences in the prakriyā part. The reasons for the differences are already discussed in the 3rd chapter. So we did not compare the prakriyā. We limited ourselves to 500 examples. Because, for most of the examples, a morphological analyser dealing with the feminine suffixes, or secondary derivative suffixes (taddhitas) was needed. In the absence of such an analyser, we could cross check our generator with only a limited cases.

Chapter 7

Conclusion

The present work is an attempt on building an Automatic Sanskrit Compound Generator. While implementing these rules we observed that most of the rules fall under the category of regular grammar. In our implementation we observed that most of the rules require a sequence of phonemes. We implemented those rules in Lex (Lexical Analyser). For other rules additional linguistic conditions such as morphological analysis, lexical meaning etc were needed and in some cases extra-linguistic conditions were required. Such rules are implemented using a C code to handle the additional linguistic conditions and an user interface to get the extra linguistic information from the user. There are around 90 rules which require additional information apart from the given input string. For the remaining rules pattern matching is enough to generate the compound.

Limitations of Sanskrit Compound Generator

The current implementation has following limitations.

- The input for the generator should be an analytical paraphrase.
- This system handles only binary compounds. Thus the generation of Dvandva with more than two components and bahupada Bahuvrīhi are not being handled in the current implementation.
- Morphological analyser is required to deal with feminine suffixes, and secondary derivative suffixes (taddhitas). Hence such cases where this information is needed are also out of scope of these current implementation.
- In case of upapadatatpuruṣa, we still do not have a morphological analyser that generates the kṛdanta forms required in the upapadatatpuruṣa compounds and hence our compound generation in such cases start with the form. For example, we start with

kumbha + ṇas kāra

and not

kumbha + ṇas kṛ + an

- We also do not deal with the accents.
- This generator produces the final output in the form of prātipadika and not a finished subanta. Since to generate a subanta, we need the feminine forms in certain cases and we still do not have a feminine form generator.

Future directions

- The above mentioned limitations need to be overcome.
- The current lexical analyser uses pattern matching only on a string. However, a lexical analyser with pattern matching on records having different fields such as the phonemic sequence, meaning, morphological analysis, genre, gender etc will be more appropriate to implement the rules in a natural way. This implementation has also given us confidence that we can, on similar lines, develop a generator for other derived forms such as *kṛt* and *taddhita*.

PART - II

Index.html

```
<html>
<head>
<meta charset="UTF-8">
<meta http-equiv="CONTENT-TYPE" content="text/html; charset=UTF-8">
<title>Samastapadavyutpaadikaa</title>

<link rel="stylesheet" type="text/css"
href="/htdocs/scl/skt_gen/new_compounds/css/compounds.css">
<script
src="http://localhost/htdocs/scl/skt_gen/new_compounds/js/jquery.min.js"></sc
ript>
<script src="http://localhost/htdocs/scl/skt_gen/new_compounds/js/jquery-
ui.js"></script>
<script
src="http://localhost/htdocs/scl/skt_gen/new_compounds/js/calcgiscripts.js"><
/script>

</head>
<body>
<center>
<div class="container-full">

<header class="head_div col-md-12 text-center">
    <h3>समस्तपदव्युत्पादिका : A compound stem generator</h3>
    <h4>Department of Sanskrit Studies, University of Hyderabad.</h4>
</header>
</div>
</center>
<div class="listdiv">
<center>
<table>
<tr>
<td>
<form name="f1" method="post">
<table>
<tr>
<td>
अधः प्रदत्तम् अलौकिकविग्रहवाक्यं चिन्वन्तु
```

</td>
 </tr>
 <tr>
 <td>
 <select name="vigraha" id="vigraha_vAkya" size="20">
 <option value="rAjan+fas puruRa+su (2.2.8)">राजन्+डस् पुरुष+सु (2.2.8)
 </option>
 <option value="mahaw+su rAjan+su (2.1.61)">महत्+सु राजन्+सु (2.1.61)
 </option>
 <option value="kudmalAgra+su xanwa+su (2.2.24)">कुड्मलाग्र+सु दन्त+सु
 (2.2.24) </option>
 <option value="nIla+su uwpala+su (2.1.57)">नील+सु उत्पल+सु
 (2.1.57)</option>
 <option value="UDa+su raWa+su (2.2.24)">ऊढ+सु रथ+सु (2.2.24)</option>
 <option value="eka+tA pari (2.1.10)">एक+टा परि (2.1.10)</option>
 <option value="ku anna+su (2.2.18)">कु अन्न+सु (2.2.18)</option>
 <option value="ku raWa+su (2.2.18)">कु रथ+सु (2.2.18)</option>
 <option value="ku vaxa+su (2.2.18)">कु वद+सु (2.2.18)</option>
 <option value="hqxaya+fas roga+su (2.2.8)">हृदय+डस् रोग+सु (2.2.8)</option>
 <option value="hqxaya+fas lAsa+su (2.2.8)">हृदय+डस् लास+सु
 (2.2.8)</option>
 <option value="hari+su hara+su (2.2.29)">हरि+सु हर+सु (2.2.29)</option>
 <option value="hala+fi wripaxikA+su (2.1.44)">हल+डि त्रिपदिका+सु
 (2.1.44)</option>
 <option value="haswa+fi banXa+su (2.1.41)">हस्त+डि बन्ध+सु
 (2.1.41)</option>
 <option value="haswin+fas varcas+su (2.2.8)">हस्तिन्+डस् वर्चस्+सु
 (2.2.8)</option>
 <option value="na karwum (2.2.6)">न कर्तुम् (2.2.6)</option>
 <option value="akeSA+su BAryA+su (2.2.24)">अकेशा+सु भार्या+सु
 (2.2.24)</option>
 <option value="akRa+fas Xur+su (2.2.8)">अक्ष+डस् धुर्+सु (2.2.8)</option>
 <option value="akRa+tA pari (2.1.10)">अक्ष+टा परि (2.1.10)</option>
 <option value="akRa+sup SONda+su (2.1.40)">अक्ष+सुप् शौण्ड+सु
 (2.1.40)</option>
 <option value="akRi+O BrU+O (2.2.29)">अक्षि+औ भ्रू+औ (2.2.29)</option>
 <option value="na ga (2.2.19)">न ग (2.2.19)</option>
 <option value="agaxa+fas kAra+su (2.2.8)">अगद+डस् कार+सु (2.2.8)</option>
 <option value="na go+su (2.2.6)">न गो+सु (2.2.6)</option>
 <option value="agni+Am swoma+su (2.2.8)">अग्नि+आम् स्तोम+सु
 (2.2.8)</option>
 <option value="agni+su varuNa+su (2.2.29)">अग्नि+सु वरुण+सु
 (2.2.29)</option>

<option value="agnIRoma+os6 CAyA+su (2.2.8)">अग्नीषोम+ओस् छाया+सु
 (2.2.8)</option>
 <option value="Ahiwa+as agni+as (2.2.24)">आहित+अस् अग्नि+अस्
 (2.2.24)</option>
 <option value="aja+fas wunxa+su (2.2.8)">अज+डस् तुन्द+सु (2.2.8)</option>
 <option value="ajAwa+su kakuxa+su (2.2.24)">अजात+सु ककुद+सु
 (2.2.24)</option>
 <option value="aNaka+su kulAla+su (2.1.54)">अणक+सु कुलाल+सु
 (2.1.54)</option>
 <option value="aNakA+su BAryA+su (2.1.54)">अणका+सु भार्या+सु
 (2.1.54)</option>
 <option value="nixrA+su awi (2.1.6)">निद्रा+सु अति (2.1.6)</option>
 <option value="hima+fas awi (2.1.6)">हिम+डस् अति (2.1.6)</option>
 <option value="kumArI+su aXi (2.1.6)">कुमारी+सु अधि (2.1.6)</option>
 <option value="gopA+fi aXi (2.1.6)">गोपा+डि अधि (2.1.6)</option>
 <option value="hari+fi aXi (2.1.6)">हरि+डि अधि (2.1.6)</option>
 <option value="Awman+fi aXi (2.1.6)">आत्मन्+डि अधि (2.1.6)</option>
 <option value="na aSva+su (2.2.6)">न अश्व+सु (2.2.6)</option>
 <option value="grAma+fas anu (2.1.16)">ग्राम+डस् अनु (2.1.16)</option>
 <option value="go+fas anu (2.1.16)">गो+डस् अनु (2.1.16)</option>
 <option value="giri+fas anu (2.1.6)">गिरि+डस् अनु (2.1.6)</option>
 <option value="giri+fas anu (2.1.6)">गिरि+डस् अनु (2.1.6)</option>
 <option value="go+fas anu (2.1.6)">गो+डस् अनु (2.1.6)</option>
 <option value="jyeRTa+fas anu (2.1.6)">ज्येष्ठ+डस् अनु (2.1.6)</option>
 <option value="na upalabXi+su (2.2.6)">न उपलब्धि+सु (2.2.6)</option>
 <option value="anu rahas+su (2.2.18)">अनु रहस्+सु (2.2.18)</option>
 <option value="rUpa+fas anu (2.1.6)">रूप+डस् अनु (2.1.6)</option>
 <option value="vana+fas anu (2.1.15)">वन+डस् अनु (2.1.15)</option>
 <option value="viRNu+fas anu (2.1.6)">विष्णु+डस् अनु (2.1.6)</option>
 <option value="giri+fi anwar (2.1.6)">गिरि+डि अन्तर् (2.1.6)</option>
 <option value="anwika+fasi Agawa+su (2.1.39)">अन्तिक+डसि आगत+सु
 (2.1.39)</option>
 <option value="anXa+su wamas+su (2.1.57)">अन्ध+सु तमस्+सु
 (2.1.57)</option>
 <option value="anya+fas kAraka+su (2.2.8)">अन्य+डस् कारक+सु
 (2.2.8)</option>
 <option value="anya+fe arWa+su (2.1.36)">अन्य+डे अर्थ+सु (2.1.36)</option>
 <option value="anya+su AsWiwa+su (2.1.57)">अन्य+सु आस्थित+सु
 (2.1.57)</option>
 <option value="anya+fas ASis+su (2.2.8)">अन्य+डस् आशिस+सु
 (2.2.8)</option>
 <option value="anya+tA AsWiwa+su (2.1.32)">अन्य+टा आस्थित+सु
 (2.1.32)</option>

<option value="akRi+os anu (2.1.6)">अक्षि+ओस् अनु (2.1.6)</option>
 <option value="anvAje+kq+wvA (1.4.73)">अन्वाजे+कृ+त्वा (1.4.73)</option>
 <option value="anvAje+kq+wvA (2.2.18)">अन्वाजे+कृ+त्वा (2.2.18)</option>
 <option value="ap+as anu (2.1.15)">अप्+अस् अनु (2.1.15)</option>
 <option value="apa kara (2.2.18)">अप कर (2.2.18)</option>
 <option value="xiSA+os apa (2.1.6)">दिशा+ओस् अप (2.1.6)</option>
 <option value="xiSA+os apa (2.1.6)">दिशा+ओस् अप (2.1.6)</option>
 <option value="na paWin+su (2.2.6)">न पथिन्+सु (2.2.6)</option>
 <option value="apara+jas para+jas (2.2.29)">अपर+जस् पर+जस्
 (2.2.29)</option>
 <option value="aparA+su BAryA+su (2.1.58)">अपरा+सु भार्या+सु
 (2.1.58)</option>
 <option value="apararAwra+fi kqwa+su (2.1.45)">अपररात्र+ङि कृत+सु
 (2.1.45)</option>
 <option value="aparA+su samA+su (2.1.17)">अपरा+सु समा+सु
 (2.1.17)</option>
 <option value="apara+su aXyApaka+su (2.1.58)">अपर+सु अध्यापक+सु
 (2.1.58)</option>
 <option value="viRNu+fas apa (2.1.12)">विष्णु+ङस् अप (2.1.12)</option>
 <option value="apUpa+fas arXa+su (2.2.8)">अपूप+ङस् अर्ध+सु (2.2.8)</option>

 <option value="ap+su cara (2.2.19)">अप्+सु चर (2.2.19)</option>
 <option value="ap+su ja (2.2.19)">अप्+सु ज (2.2.19)</option>
 <option value="ap+su mawi+su (2.2.24)">अप्+सु मति+सु (2.2.24)</option>
 <option value="ap+sup yoni+su (2.2.24)">अप्+सुप् योनि+सु (2.2.24)</option>
 <option value="na brAhmana+su (2.2.6)">न ब्राह्मन्+सु (2.2.6)</option>
 <option value="agni+am aBi (2.1.14)">अग्नि+अम् अभि (2.1.14)</option>
 <option value="aByASa+fasi Agawa+su (2.1.39)">अभ्याश+ङसि आगत+सु
 (2.1.39)</option>
 <option value="amqwa+su aSman+su (2.1.57)">अमृत+सु अश्मन्+सु
 (2.1.57)</option>
 <option value="ayas+su xanwa+as (2.2.24)">अयस्+सु दन्त+अस्
 (2.2.24)</option>
 <option value="araNya+am gawa+su (2.1.24)">अरण्य+अम् गत+सु
 (2.1.24)</option>
 <option value="araNya+fi wilaka+as (2.1.44)">अरण्य+ङि तिलक+अस्
 (2.1.44)</option>
 <option value="na rAjan+su (2.2.6)">न राजन्+सु (2.2.6)</option>
 <option value="Siras+fi arus+su (2.2.24)">शिरस्+ङि अरुस्+सु (2.2.24)</option>

 <option value="aroka+jas xanwa+jas (2.2.24)">अरोक+जस् दन्त+जस्
 (2.2.24)</option>
 <option value="arka+su aSvameXa+su (2.2.29)">अर्क+सु अश्वमेध+सु

(2.2.29)</option>
 <option value="arWa+fas gOrava+su (2.2.8)">अर्थ+डस् गौरव+सु
 (2.2.8)</option>
 <option value="arWa+su Xarma+su (2.2.29)">अर्थ+सु धर्म+सु (2.2.29)</option>
 <option value="KArI+fas arXa+su (2.2.2)">खारी+डस् अर्थ+सु (2.2.2)</option>
 <option value="arXa+su wqwIya+su (2.2.24)">अर्थ+सु तृतीय+सु
 (2.2.24)</option>
 <option value="nO+fas arXa+su (2.2.2)">नौ+डस् अर्थ+सु (2.2.2)</option>
 <option value="rAwri+fas arXa+su (2.2.2)">रात्रि+डस् अर्थ+सु (2.2.2)</option>
 <option value="qc+fas arXa+su (2.2.2)">ऋच्+डस् अर्थ+सु (2.2.2)</option>
 <option value="kumArI+fe alam (2.2.18)">कुमारी+डे अलम् (2.2.18)</option>
 <option value="alam kqwa+su (1.4.64)">अलम् कृत+सु (1.4.64)</option>
 <option value="alam kqwa+su (2.2.18)">अलम् कृत+सु (2.2.18)</option>
 <option value="alAbukarkanXuxqnBU+Am Pala+su
 (2.2.8)">अलाबुकर्क-धुदृन्भू+आम् फल+सु (2.2.8)</option>
 <option value="alpA+su meXA+su (2.2.24)">अल्पा+सु मेधा+सु
 (2.2.24)</option>
 <option value="alpa+fasi mukwa+su (2.1.39)">अल्प+डसि मुक्त+सु
 (2.1.39)</option>
 <option value="ava kara (2.2.18)">अव कर (2.2.18)</option>
 <option value="pakva+su avaklinna+su (2.1.49)">पक्व+सु अवक्लिन्न+सु
 (2.1.49)</option>
 <option value="avata+i kacCapa+su (2.1.48)">अवट+इ कच्छप+सु
 (2.1.48)</option>
 <option value="avawapwa+fi nakulasWiwa+su (2.1.47)">अवतप्त+डि
 नकुलस्थित+सु (2.1.47)</option>
 <option value="ava wamas+su (2.2.18)">अव तमस्+सु (2.2.18)</option>
 <option value="avamukwA+O upAnah+O (2.2.24)">अवमुक्ता+औ उपानह्+औ
 (2.2.24)</option>
 <option value="ava rahas+su (2.2.18)">अव रहस्+सु (2.2.18)</option>
 <option value="avaSyam BAvyA+su (2.1.72)">अवश्यम् भाव्य+सु
 (2.1.72)</option>
 <option value="aSva+tA krIwa+su (2.1.32)">अश्व+टा क्रीत+सु (2.1.32)</option>
 <option value="aSva+Am GAsa+su (2.2.8)">अश्व+आम् घास+सु
 (2.2.8)</option>
 <option value="aSva+jas vadavA+jas (2.2.29)">अश्व+जस् वडवा+जस्
 (2.2.29)</option>
 <option value="aSva+Am uras+su (2.2.8)">अश्व+आम् उरस्+सु (2.2.8)</option>
 <option value="aRtan+jas cawvAriMSaw+su (2.2.29)">अष्टन्+जस् चत्वारिंशत्+सु
 (2.2.29)</option>
 <option value="aRtan+jas puwra+jas (2.2.24)">अष्टन्+जस् पुत्र+जस्
 (2.2.24)</option>

<option value="aRtan+jas kapAla+jas (2.1.51)">अष्टन्+जस् कपाल+जस्
 (2.1.51)</option>
 <option value="aRtan+jas go+jas (2.2.24)">अष्टन्+जस् गो+जस् (2.2.24)</option>
 <option value="aRtan+jas paxa+as (2.2.24)">अष्टन्+जस् पद+अस्
 (2.2.24)</option>
 <option value="aRtan+jas viMSawi+su (2.2.29)">अष्टन्+जस् विंशति+सु
 (2.2.29)</option>
 <option value="na saKi+su (2.2.6)">न सखि+सु (2.2.6)</option>
 <option value="asaw kqwvA (1.4.63)">असत् कृत्वा (1.4.63)</option>
 <option value="asaw kqwvA (2.2.18)">असत् कृत्वा (2.2.18)</option>
 <option value="asipawra+Am vana+su (2.2.8)">असिपत्र+आम् वन+सु
 (2.2.8)</option>
 <option value="na sena+su (2.2.6)">न सेन+सु (2.2.6)</option>
 <option value="aswam gam+wvA (1.4.68)">अस्तम् गम्+त्वा (1.4.68)</option>
 <option value="aswam gam+wvA (2.2.18)">अस्तम् गम्+त्वा (2.2.18)</option>
 <option value="aswui kAra+su (2.1.4)">अस्तुइ कार+सु (2.1.4)</option>
 <option value="uxyawa+su asi+su (2.2.24)">उद्यत+सु असि+सु (2.2.24)</option>
 <option value="ahan+Am safkrAnwa+as (2.1.28)">अहन्+आम् सङ्क्रान्त+अस्
 (2.1.28)</option>
 <option value="ahan+fi xivA+su (2.2.29)">अहन्+ङि दिवा+सु (2.2.29)</option>
 <option value="ahi+jas nakula+jas (2.2.29)">अहि+जस् नकुल+जस्
 (2.2.29)</option>
 <option value="ahan+su rAwri+su (2.2.29)">अहन्+सु रात्रि+सु (2.2.29)</option>
 <option value="A kadAra+su (2.2.18)">आ कडार+सु (2.2.18)</option>
 <option value="AkarRa+su Svan+su (2.1.56)">आकर्ष+सु श्वन्+सु
 (2.1.56)</option>
 <option value="AKara+fi sWa (2.2.19)">आखर+ङि स्थ (2.2.19)</option>
 <option value="agni+su viRNu+su (2.2.29)">अग्नि+सु विष्णु+सु (2.2.29)</option>
 <option value="agni+su maruw+su (2.2.29)">अग्नि+सु मरुत्+सु (2.2.29)</option>
 <option value="agni+su varuna+su (2.2.29)">अग्नि+सु वरुन+सु (2.2.29)</option>
 <option value="agni+su inxra+su (2.2.29)">अग्नि+सु इन्द्र+सु (2.2.29)</option>
 <option value="A carya (2.2.18)">आ चर्य (2.2.18)</option>
 <option value="AcAra+tA nipuna+su (2.1.31)">आचार+टा निपुन+सु
 (2.1.31)</option>
 <option value="Awapa+fi SuRka+su (2.1.41)">आतप+ङि शुष्क+सु
 (2.1.41)</option>
 <option value="Awman+tA kqwa+su (2.1.32)">आत्मन्+टा कृत+सु
 (2.1.32)</option>

<option value="Awman+su cawurWa+su (2.2.24)">आत्मन्+सु चतुर्थ+सु
 (2.2.24)</option>
 <option value="Awman+tA paFcama+su (2.1.30)">आत्मन्+टा पञ्चम+सु
 (2.1.30)</option>
 <option value="Awman+fe paxa+su (2.1.36)">आत्मन्+डे पद+सु
 (2.1.36)</option>
 <option value="A paxam+fasi (2.1.13)">आ पदम्+डसि (2.1.13)</option>
 <option value="Apanna+su jIvikA+am (2.2.4)">आपन्न+सु जीविका+अम्
 (2.2.4)</option>
 <option value="aparA+su SAIA+su (2.1.51)">अपरा+सु शाला+सु
 (2.1.51)</option>
 <option value="A baxXa+su (2.2.18)">आ बद्ध+सु (2.2.18)</option>
 <option value="bAla+Byas A (2.1.13)">बाल+भ्यस् आ (2.1.13)</option>
 <option value="mukwi+fasi A (2.1.13)">मुक्ति+डसि आ (2.1.13)</option>
 <option value="axas+fas kula+su (2.2.8)">अदस्+डस् कुल+सु (2.2.8)</option>
 <option value="Ayus+Am swoma+su (2.2.8)">आयुस्+आम् स्तोम+सु
 (2.2.8)</option>
 <option value="Arxra+su gomaya+su (2.1.57)">आर्द्र+सु गोमय+सु
 (2.1.57)</option>
 <option value="Ahiwa+jas agni+jas (2.2.24)">आहित+जस् अग्नि+जस्
 (2.2.24)</option>
 <option value="ikRu+Am CAyA+su (2.2.8)">इक्षु+आम् छाया+सु
 (2.2.8)</option>
 <option value="ikRu+Am BakRikA+su (2.2.8)">इक्षु+आम् भक्षिका+सु
 (2.2.8)</option>
 <option value="ikRu+Am vAhana+su (2.2.8)">इक्षु+आम् वाहन+सु
 (2.2.8)</option>
 <option value="hari+fas iwi (2.1.6)">हरि+डस् इति (2.1.6)</option>
 <option value="iXma+su barhis+su (2.2.29)">इध्म+सु बर्हिस्+सु
 (2.2.29)</option>
 <option value="ina+fas saBA+su (2.2.8)">इन+डस् सभा+सु (2.2.8)</option>
 <option value="inxra+fas vAhana+su (2.2.8)">इन्द्र+डस् वाहन+सु
 (2.2.8)</option>
 <option value="inxra+su agni+su (2.2.29)">इन्द्र+सु अग्नि+सु (2.2.29)</option>
 <option value="potA+su iBI+su (2.1.65)">पोटा+सु इभी+सु (2.1.65)</option>
 <option value="yuvawi+su iBI+su (2.1.65)">युवति+सु इभी+सु (2.1.65)</option>
 <option value="irikA+Am vana+su (2.2.8)">इरिका+आम् वन+सु (2.2.8)</option>
 <option value="iRIkA+fas wUla+su (2.2.8)">इषीका+डस् तूल+सु
 (2.2.8)</option>
 <option value="iRu+su vajra+su (2.2.29)">इषु+सु वज्र+सु (2.2.29)</option>
 <option value="iRtakA+Bis ciwa+su (2.1.32)">इष्टका+भिस् चित+सु
 (2.1.32)</option>

<option value="ixam+as xqS (2.2.19)">इदम्+अस् दृश् (2.2.19)</option>
 <option value="ISa+su kqRNa+su (2.2.29)">ईश+सु कृष्ण+सु (2.2.29)</option>
 <option value="ISvara+fas saBA+su (2.2.8)">ईश्वर+डस् सभा+सु
 (2.2.8)</option>
 <option value="ISvara+fi aXi (2.1.40)">ईश्वर+ङि अधि (2.1.40)</option>
 <option value="IRaw pifgala+su (2.2.7)">ईषत् पिङ्गल+सु (2.2.7)</option>
 <option value="ugra+fas xuhiwq+su (2.2.8)">उग्र+डस् दुहितृ+सु (2.2.8)</option>

 <option value="uccA+su Xur+su (2.2.24)">उच्चा+सु धुर्+सु (2.2.24)</option>
 <option value="uxak avAk (2.1.72)">उदक् अवाक् (2.1.72)</option>
 <option value="uwkqRta+su puruRa+su (2.1.61)">उत्कृष्ट+सु पुरुष+सु
 (2.1.61)</option>
 <option value="uwwama+su go+su (2.1.61)">उत्तम+सु गो+सु (2.1.61)</option>
 <option value="uwwamA+su Xur+su (2.1.61)">उत्तमा+सु धुर्+सु
 (2.1.61)</option>
 <option value="uwwama+su ahan+su (2.1.61)">उत्तम+सु अहन्+सु
 (2.1.61)</option>
 <option value="uwwara+su wIra+su (2.1.57)">उत्तर+सु तीर+सु
 (2.1.57)</option>
 <option value="uwwara+su sakWi+fas (2.2.1)">उत्तर+सु सक्थि+डस्
 (2.2.1)</option>
 <option value="uxaka+fas kumBa+su (2.2.8)">उदक+डस् कुम्भ+सु
 (2.2.8)</option>
 <option value="uxaka+fas gAha (2.2.19)">उदक+डस् गाह (2.2.19)</option>
 <option value="uxaka+fas parvawa+su (2.2.8)">उदक+डस् पर्वत+सु
 (2.2.8)</option>
 <option value="uxaka+fas binxu+su (2.2.8)">उदक+डस् बिन्दु+सु
 (2.2.8)</option>
 <option value="uxaka+tA manWa+su (2.1.30)">उदक+टा मन्थ+सु
 (2.1.30)</option>
 <option value="uxaka+fas sWAlI+su (2.2.8)">उदक+डस् स्थाली+सु
 (2.2.8)</option>
 <option value="uxaka+fas kumBa+su (2.2.8)">उदक+डस् कुम्भ+सु
 (2.2.8)</option>
 <option value="uxaka+tA oxana+su (2.1.34)">उदक+टा ओदन+सु
 (2.1.34)</option>
 <option value="uxIcI+su BUmi+su (2.2.24)">उदीची+सु भूमि+सु
 (2.2.24)</option>
 <option value="uxaka+Sas Xi (2.2.19)">उदक+शस् धि (2.2.19)</option>
 <option value="uxaka+tA peRam (2.2.20)">उदक+टा पेषम् (2.2.20)</option>
 <option value="uxaka+fas binxu+su (2.2.8)">उदक+डस् बिन्दु+सु
 (2.2.8)</option>
 <option value="uxaka+fas meGa+su (2.2.8)">उदक+डस् मेघ+सु (2.2.8)</option>

<option value="uxaka+fas vAsa+su (2.2.8)">उदक+डस् वास+सु (2.2.8)</option>

<option value="uxaka+fas vAhana+su (2.2.8)">उदक+डस् वाहन+सु (2.2.8)</option>

<option value="kawipaya+su uxagAw+su (2.1.65)">कतिपय+सु उदगात्+सु (2.1.65)</option>

<option value="uxaka+tA oxana+su (2.1.34)">उदक+टा ओदन+सु (2.1.34)</option>

<option value="uxxAlakapuRpa+Am BaFjikA+su (2.2.17)">उद्दालकपुष्प+आम् भञ्जिका+सु (2.2.17)</option>

<option value="uxXqwa+su oxana+su (2.2.24)">उद्धृत+सु ओदन+सु (2.2.24)</option>

<option value="uxXya+su irAvawI+su (2.2.29)">उद्धय+सु इरावती+सु (2.2.29)</option>

<option value="unmawwA+su gafgA+su (2.1.21)">उन्मत्ता+सु गङ्गा+सु (2.1.21)</option>

<option value="kqRNa+as upa (2.1.6)">कृष्ण+अस् उप (2.1.6)</option>

<option value="gana+Am upa (2.2.25)">गन+आम् उप (2.2.25)</option>

<option value="giri+fas upa (2.1.6)">गिरि+डस् उप (2.1.6)</option>

<option value="cawur+Am upa (2.2.25)">चतुर्+आम् उप (2.2.25)</option>

<option value="carman+fas upa (2.1.6)">चर्मन्+डस् उप (2.1.6)</option>

<option value="jarA+fas upa (2.1.6)">जरा+डस् उप (2.1.6)</option>

<option value="naxI+fas upa (2.1.6)">नदी+डस् उप (2.1.6)</option>

<option value="pOrNamAsI+fas upa (2.1.6)">पौर्णमासी+डस् उप (2.1.6)</option>

<option value="bahu+Am upa (2.2.25)">बहु+आम् उप (2.2.25)</option>

<option value="rAjan+fas upa (2.1.6)">राजन्+डस् उप (2.1.6)</option>

<option value="Sarax+fas upa (2.1.6)">शरद्+डस् उप (2.1.6)</option>

<option value="Svan+fas upa (2.1.6)">श्वन्+डस् उप (2.1.6)</option>

<option value="samiX+fas upa (2.1.6)">समिध्+डस् उप (2.1.6)</option>

<option value="upasara+fi ja (2.2.19)">उपसर+ङि ज (2.2.19)</option>

<option value="upahqwa+su pasu+su (2.2.24)">उपहत+सु पसु+सु (2.2.24)</option>

<option value="AgrahAyaNI+fas upa (2.1.6)">आग्रहायणी+डस् उप (2.1.6)</option>

<option value="uBa+O haswa+O (2.2.24)">उभ+औ हस्त+औ (2.2.24)</option>

<option value="uBa+O aFjali+O (2.2.24)">उभ+औ अञ्जलि+औ (2.2.24)</option>

<option value="uras+fi loman+jas (2.2.24)">उरस्+ङि लोमन्+जस् (2.2.24)</option>

<option value="uRas+su nakwam (2.2.29)">उषस्+सु नक्तम् (2.2.29)</option>

<option value="uRas+su sUrya+su (2.2.29)">उषस्+सु सूर्य+सु (2.2.29)</option>

<option value="uRNa+fas karana+su (2.2.8)">उष्ण+डस् करन+सु
 (2.2.8)</option>
 <option value="UDa+su raWa+su (2.2.24)">ऊढ+सु रथ+सु (2.2.24)</option>
 <option value="UrI kqwa+su (2.2.18)">ऊरी कृत+सु (2.2.18)</option>
 <option value="UrXva+O jAnu+O (2.2.24)">ऊर्ध्व+औ जानु+औ
 (2.2.24)</option>
 <option value="Uru+O aRTIvaw+O (2.2.29)">ऊरु+औ अष्टीवत्+औ
 (2.2.29)</option>
 <option value="qc+su sAman+su (2.2.29)">ऋच्+सु सामन्+सु (2.2.29)</option>
 <option value="qc+Am ayana+su (2.2.8)">ऋच्+आम् अयन+सु (2.2.8)</option>
 <option value="qc+su yajus+su (2.2.29)">ऋच्+सु यजुस्+सु (2.2.29)</option>
 <option value="qRi+Am vaha+su (2.2.8)">ऋषि+आम् वह+सु (2.2.8)</option>
 <option value="ekA+fas kRIra+su (2.2.8)">एका+डस् क्षीर+सु (2.2.8)</option>
 <option value="ekA+su ciwi+su (2.2.24)">एका+सु चिति+सु (2.2.24)</option>
 <option value="eka+su nAWa+su (2.1.49)">एक+सु नाथ+सु (2.1.49)</option>
 <option value="eka+tA pari (2.1.10)">एक+टा परि (2.1.10)</option>
 <option value="eka+su viMSawi+su (2.2.29)">एक+सु विंशति+सु
 (2.2.29)</option>
 <option value="eka+su xaSan+as (2.2.29)">एक+सु दशन्+अस् (2.2.29)</option>
 <option value="na viMSawi+su (2.2.6)">न विंशति+सु (2.2.6)</option>
 <option value="eka+tA naviMSawi+su (2.1.30)">एक+टा नविंशति+सु
 (2.1.30)</option>
 <option value="eka+su ahan+su (2.1.49)">एक+सु अहन्+सु (2.1.49)</option>
 <option value="ojas+tA kqwa+su (2.1.32)">ओजस्+टा कृत+सु (2.1.32)</option>

 <option value="kacCa+tA pa (2.2.19)">कच्छ+टा प (2.2.19)</option>
 <option value="kata+jas prU (2.2.19)">कट+जस् प्रू (2.2.19)</option>
 <option value="kaTa+jas kAlApa+jas (2.2.29)">कठ+जस् कालाप+जस्
 (2.2.29)</option>
 <option value="kadAra+su jEmini+su (2.1.57)">कडार+सु जैमिनि+सु
 (2.1.57)</option>
 <option value="kawama+su kAlApa+su (2.1.63)">कतम+सु कालाप+सु
 (2.1.63)</option>
 <option value="kawara+su kaTa+su (2.1.63)">कतर+सु कठ+सु
 (2.1.63)</option>
 <option value="ku wqNa+su (2.2.18)">कु तृण+सु (2.2.18)</option>
 <option value="ku wri+jas (2.2.18)">कु त्रि+जस् (2.2.18)</option>
 <option value="ku anna+su (2.2.18)">कु अन्न+सु (2.2.18)</option>
 <option value="ku raWa+su (2.2.18)">कु रथ+सु (2.2.18)</option>
 <option value="ku vaxa+su (2.2.18)">कु वद+सु (2.2.18)</option>
 <option value="kabara+su akRi+su (2.1.56)">कबर+सु अक्षि+सु
 (2.1.56)</option>
 <option value="karna+fi japa+su (2.2.19)">कर्न+ङि जप+सु (2.2.19)</option>

<option value="kalpana+fasi apoDa+su (2.1.38)">कल्पन+डसि अपोढ+सु
 (2.1.38)</option>
 <option value="kalyAna+su Xarma+su (2.2.24)">कल्यान+सु धर्म+सु
 (2.2.24)</option>
 <option value="kalyANI+su paFcaml+su (2.2.24)">कल्याणी+सु पञ्चमी+सु
 (2.2.24)</option>
 <option value="kalyAna+O akRi+O (2.2.24)">कल्यान+औ अक्षि+औ
 (2.2.24)</option>
 <option value="kalyANI+su praXAna+su (2.2.24)">कल्याणी+सु प्रधान+सु
 (2.2.24)</option>
 <option value="kalyANI+su priyA+su (2.2.24)">कल्याणी+सु प्रिया+सु
 (2.2.24)</option>
 <option value="kalyANI+fas mAwq+su (2.2.8)">कल्याणी+डस् मातृ+सु
 (2.2.8)</option>
 <option value="ku uRNa+su (2.2.18)">कु उष्ण+सु (2.2.18)</option>
 <option value="kAka+Bis peyA+su (2.1.33)">काक+भिस् पेया+सु
 (2.1.33)</option>
 <option value="kAkI+fas SAva+su (2.2.8)">काकी+डस् शाव+सु (2.2.8)</option>

 <option value="ku akRi+O (2.2.24)">कु अक्षि+औ (2.2.24)</option>
 <option value="ku akRa+su (2.2.18)">कु अक्ष+सु (2.2.18)</option>
 <option value="ku jala+su (2.2.18)">कु जल+सु (2.2.18)</option>
 <option value="kANdalAvI+su BAryA+su (2.2.24)">काण्डलावी+सु भार्या+सु
 (2.2.24)</option>
 <option value="ku wIra+su (2.2.18)">कु तीर+सु (2.2.18)</option>
 <option value="ku paWin+su (2.2.18)">कु पथिन्+सु (2.2.18)</option>
 <option value="ku amla+su (2.2.18)">कु अम्ल+सु (2.2.18)</option>
 <option value="kAra+jas kara (2.2.19)">कार+जस् कर (2.2.19)</option>
 <option value="kArikA+su vqnxArikA+su (2.1.62)">कारिका+सु वृन्दारिका+सु
 (2.1.62)</option>
 <option value="kArIRaganXyA+su banXu+su (2.2.24)">कारीषगन्ध्या+सु बन्धु+सु
 (2.2.24)</option>
 <option value="kArIRaganXyA+su mAwq+su (2.2.24)">कारीषगन्ध्या+सु मातृ+सु
 (2.2.24)</option>
 <option value="kArIRaganXyA+fas kula+su (2.2.8)">कारीषगन्ध्या+डस् कुल+सु
 (2.2.8)</option>
 <option value="kArIRaganXyA+su pawI+su (2.2.24)">कारीषगन्ध्या+सु पति+सु
 (2.2.24)</option>
 <option value="kArIRaganXyA+fas pawikula+su (2.2.8)">कारीषगन्ध्या+डस्
 पतिकुल+सु (2.2.8)</option>
 <option value="kArIRaganXyA+fas paramapawi+su (2.2.8)">कारीषगन्ध्या+डस्
 परमपति+सु (2.2.8)</option>
 <option value="kArIRaganXyA+fas paramapuwra+su (2.2.8)">कारीषगन्ध्या+डस्

परमपुत्र+सु (2.2.8)</option>
 <option value="kArIRaganXyA+su paramabanXu+su (2.2.24)">कारीषगन्ध्या+सु
 परमबन्धु+सु (2.2.24)</option>
 <option value="kArIRaganXyA+fas puwrakula+su (2.2.8)">कारीषगन्ध्या+डस्
 पुत्रकुल+सु (2.2.8)</option>
 <option value="kArIRaganXyA+fas banXu+su (2.2.8)">कारीषगन्ध्या+डस्
 बन्धु+सु (2.2.8)</option>
 <option value="kArIRaganXyAbanXu+su Xana+su
 (2.2.24)">कारीषगन्ध्याबन्धु+सु धन+सु (2.2.24)</option>
 <option value="kArRya+Am vana+su (2.2.8)">कार्ष्य+आम् वन+सु
 (2.2.8)</option>
 <option value="kAla+su ayas+su (2.1.57)">काल+सु अयस्+सु (2.1.57)</option>
 <option value="kAla+fi ja (2.2.19)">काल+डि ज (2.2.19)</option>
 <option value="kASa+Am katIkAra+su (2.2.8)">काश+आम् कटीकार+सु
 (2.2.8)</option>
 <option value="kARAyI+su kanWA+su (2.2.24)">काषायी+सु कन्था+सु
 (2.2.24)</option>
 <option value="kARTa+Am saBA+su (2.2.8)">काष्ठ+आम् सभा+सु
 (2.2.8)</option>
 <option value="ku wIra+su (2.2.24)">कु तीर+सु (2.2.24)</option>
 <option value="kim+fas go+su (2.2.8)">किम्+डस् गो+सु (2.2.8)</option>
 <option value="kim+su go+su (2.1.64)">किम्+सु गो+सु (2.1.64)</option>
 <option value="kim+fas rAjan+su (2.2.8)">किम्+डस् राजन्+सु (2.2.8)</option>
 <option value="kim+su rAjan+su (2.1.64)">किम्+सु राजन्+सु (2.1.64)</option>
 <option value="kiMSuluka+su giri+su (2.1.57)">किशुलुक+सु गिरि+सु
 (2.1.57)</option>
 <option value="kim+fas saKi+su (2.2.8)">किम्+डस् सखि+सु (2.2.8)</option>
 <option value="kim+su saKi+su (2.1.64)">किम्+सु सखि+सु (2.1.64)</option>
 <option value="kim+su sabrahmacArin+su (2.2.24)">किम्+सु सब्रह्मचारिन्+सु
 (2.2.24)</option>
 <option value="kim+jas Xa (2.2.19)">किम्+जस् ध (2.2.19)</option>
 <option value="kim+jas xqS (2.2.19)">किम्+जस् दृश् (2.2.19)</option>
 <option value="kukkuta+su mayUrI+su (2.2.29)">कुक्कुट+सु मयूरी+सु
 (2.2.29)</option>
 <option value="kudmalAgra+jas xanwa+jas (2.2.24)">कुड्मलाग्र+जस् दन्त+जस्
 (2.2.24)</option>
 <option value="kudmalAgra+jas xanwa+jas (2.2.24)">कुड्मलाग्र+जस् दन्त+जस्
 (2.2.24)</option>
 <option value="kudya+fas CAyA+su (2.2.8)">कुड्य+डस् छाया+सु
 (2.2.8)</option>
 <option value="kudya+fas CAyA+su (2.2.8)">कुड्य+डस् छाया+सु
 (2.2.8)</option>
 <option value="ku wumburu+jas (2.2.18)">कु तुम्बुरु+जस् (2.2.18)</option>

<option value="ku wqna+jas (2.2.18)">कु तृन+जस् (2.2.18)</option>
 <option value="ku puruRa+su (2.2.18)">कु पुरुष+सु (2.2.18)</option>
 <option value="kubja+su KaFja+su (2.1.57)">कुब्ज+सु खञ्ज+सु </option>
 <option value="ku brahman+su (2.2.18)">कु ब्रह्मन्+सु (2.2.18)</option>
 <option value="ku brahman+su (2.2.18)">कु ब्रह्मन्+सु (2.2.18)</option>
 <option value="kumArI+jas xa (2.2.19)">कुमारी+जस् द (2.2.19)</option>
 <option value="kumArI+su mawA+su (2.1.53)">कुमारी+सु मता+सु
 (2.1.53)</option>
 <option value="kumuxa+su SyenI+su (2.1.55)">कुमुद+सु श्येनी+सु
 (2.1.55)</option>
 <option value="kumBa+jas kAra (2.2.19)">कुम्भ+जस् कार (2.2.19)</option>
 <option value="kumBa+su pAxa+su (2.2.24)">कुम्भ+सु पाद+सु
 (2.2.24)</option>
 <option value="kuru+jas kurukRewra+su (2.2.29)">कुरु+जस् कुरुक्षेत्र+सु
 (2.2.29)</option>
 <option value="kuru+su cara (2.2.19)">कुरु+सु चर (2.2.19)</option>
 <option value="kuSa+jas kASa+jas (2.2.29)">कुश+जस् काश+जस्
 (2.2.29)</option>
 <option value="kuSa+as kASa+as (2.2.29)">कुश+अस् काश+अस्
 (2.2.29)</option>
 <option value="ku uRtra+su (2.2.24)">कु उष्ट्र+सु (2.2.24)</option>
 <option value="kqwa+su akqwa+su (2.1.60)">कृत+सु अकृत+सु
 (2.1.60)</option>
 <option value="kqwa+su akqwa+su (2.2.29)">कृत+सु अकृत+सु
 (2.2.29)</option>
 <option value="kqwa+su arWa+su (2.2.24)">कृत+सु अर्थ+सु (2.2.24)</option>
 <option value="kqwwikA+su rohiNI+su (2.2.29)">कृत्तिका+सु रोहिणी+सु
 (2.2.29)</option>
 <option value="kqRNA+su cawurxaSI+su (2.1.57)">कृष्ण+सु चतुर्दशी+सु
 (2.1.57)</option>
 <option value="kqRNA+jas wila+jas (2.1.57)">कृष्ण+जस् तिल+जस् </option>
 <option value="kqRNA+su BUmi+su (2.2.24)">कृष्ण+सु भूमि+सु
 (2.2.24)</option>
 <option value="kqRNA+am Sriwa+su (2.1.24)">कृष्ण+अम् श्रित+सु
 (2.1.24)</option>
 <option value="kqRNA+fas saKi+su (2.2.8)">कृष्ण+डस् सखि+सु
 (2.2.8)</option>
 <option value="kqRNA+su sarpa+su (2.1.57)">कृष्ण+सु सर्प+सु
 (2.1.57)</option>
 <option value="kqRNA+su sArafga+su (2.1.69)">कृष्ण+सु सारङ्ग+सु
 (2.1.69)</option>
 <option value="kevala+jas vEyAkarana+jas (2.1.49)">केवल+जस् वैयाकरण+जस्
 (2.1.49)</option>

<option value="keSa+su SmaSru+su (2.2.29)">केश+सु श्मश्रु+सु
 (2.2.29)</option>
 <option value="keSa+su keSa+su (2.2.27)">केश+सु केश+सु (2.2.27)</option>
 <option value="kElAsa+su ganXamAxana+su (2.2.29)">कैलास+सु गन्धमादन+सु
 (2.2.29)</option>
 <option value="kotara+Am vana+su (2.2.8)">कोटर+आम् वन+सु
 (2.2.8)</option>
 <option value="koSa+su niRax+su (2.2.29)">कोश+सु निषद्+सु
 (2.2.29)</option>
 <option value="ku uRna+su (2.2.18)">कु उष्ण+सु (2.2.18)</option>
 <option value="kOta+su wakRan+su (2.1.57)">कौट+सु तक्षन्+सु
 (2.1.57)</option>
 <option value="kOmuxaganXyA+fas pawī+su (2.2.8)">कौमुदगन्ध्या+डस् पति+सु
 (2.2.8)</option>
 <option value="kOmuxaganXyA+su banXu+su (2.2.24)">कौमुदगन्ध्या+सु
 बन्धु+सु (2.2.24)</option>
 <option value="kOmuxaganXyA+fas kula+su (2.2.8)">कौमुदगन्ध्या+डस् कुल+सु
 (2.2.8)</option>
 <option value="kRara+fi ja (2.2.19)">क्षर+ङि ज (2.2.19)</option>
 <option value="kRIra+fas pAna+su (2.2.8)">क्षीर+डस् पान+सु (2.2.8)</option>
 <option value="kRIra+su pAna+su (2.2.24)">क्षीर+सु पान+सु (2.2.24)</option>
 <option value="kRIra+am pa (2.2.19)">क्षीर+अम् प (2.2.19)</option>
 <option value="kRIra+fas pAna+su (2.2.8)">क्षीर+डस् पान+सु (2.2.8)</option>
 <option value="kRIra+su uxaka+su (2.2.24)">क्षीर+सु उदक+सु
 (2.2.24)</option>
 <option value="KaFja+su kubja+su (2.1.57)">खञ्ज+सु कुब्ज+सु
 (2.1.57)</option>
 <option value="KatvA+fas pAxa+su (2.2.8)">खट्वा+डस् पाद+सु (2.2.8)</option>

 <option value="KatvA+Am ArUDa+su (2.1.26)">खट्वा+आम् आरूढ+सु
 (2.1.26)</option>
 <option value="KatvA+su vqnxArikA+su (2.1.62)">खट्वा+सु वृन्दारिका+सु
 (2.1.62)</option>
 <option value="KarA+su nAsikA+su (2.2.24)">खरा+सु नासिका+सु
 (2.2.24)</option>
 <option value="Kara+Am yuga+as (2.2.8)">खर+आम् युग+अस् (2.2.8)</option>
 <option value="Ka+fi Saya (2.2.19)">ख+ङि शय (2.2.19)</option>
 <option value="Kura+su nAsikA+su (2.2.24)">खुर+सु नासिका+सु
 (2.2.24)</option>
 <option value="gafgA+fas pAra+su (2.2.8)">गङ्गा+डस् पार+सु (2.2.8)</option>
 <option value="gafgA+su BAryA+su (2.2.24)">गङ्गा+सु भार्या+सु
 (2.2.24)</option>
 <option value="gafgA+su yamuna+su (2.2.29)">गङ्गा+सु यमुन+सु

(2.2.29)</option>
 <option value="gafgA+su Sona+su (2.2.29)">गङ्गा+सु शोन+सु (2.2.29)</option>
 <option value="kaNTa+fi gadu+su (2.2.24)">कण्ठ+ङि गडु+सु (2.2.24)</option>
 <option value="gana+Am rAwri+Am (2.1.51)">गन+आम् रात्रि+आम्
 (2.1.51)</option>
 <option value="gamana+su AkuFcana+su (2.2.29)">गमन+सु आकुञ्चन+सु
 (2.2.29)</option>
 <option value="garga+su gArgyAyana+su (2.2.29)">गर्ग+सु गार्ग्यायन+सु
 (2.2.29)</option>
 <option value="garga+su gArgya+su (2.2.29)">गर्ग+सु गार्ग्य+सु (2.2.29)</option>

 <option value="garga+Am Baga+su (2.2.8)">गर्ग+आम् भग+सु (2.2.8)</option>
 <option value="garga+Am BaginI+su (2.2.8)">गर्ग+आम् भगिनी+सु
 (2.2.8)</option>
 <option value="garga+Am vAhana+su (2.2.8)">गर्ग+आम् वाहन+सु
 (2.2.8)</option>
 <option value="garBiNI+su BAryA+su (2.2.24)">गर्भिणी+सु भार्या+सु
 (2.2.24)</option>
 <option value="go+Am akRi+su (2.2.8)">गो+आम् अक्षि+सु (2.2.8)</option>
 <option value="go+as aSva+as (2.2.29)">गो+अस् अश्व+अस् (2.2.29)</option>
 <option value="go+fi sWira+su (2.1.44)">गो+ङि स्थिर+सु (2.1.44)</option>
 <option value="uxGa+su go+su (2.1.66)">उद्ध+सु गो+सु (2.1.66)</option>
 <option value="gArgya+su vAwsyAyana+su (2.2.29)">गार्ग्य+सु वात्स्यायन+सु
 (2.2.29)</option>
 <option value="giri+fas naxI+su (2.2.8)">गिरि+डस् नदी+सु (2.2.8)</option>
 <option value="gila+jas gila (2.2.19)">गिल+जस् गिल (2.2.19)</option>
 <option value="guda+tA XAna+as (2.1.35)">गुड+टा धान+अस् (2.1.35)</option>

 <option value="priya+su guda+su (2.2.24)">प्रिय+सु गुड+सु (2.2.24)</option>
 <option value="gupwi+fi banXa+su (2.1.44)">गुप्ति+ङि बन्ध+सु
 (2.1.44)</option>
 <option value="gUDa+su Awman+su (2.1.57)">गूढ+सु आत्मन्+सु
 (2.1.57)</option>
 <option value="geha+fi narxin+su (2.1.48)">गेह+ङि नर्दिन्+सु (2.1.48)</option>
 <option value="go+Am kula+su (2.2.8)">गो+आम् कुल+सु (2.2.8)</option>
 <option value="garBiNI+su go+su (2.1.71)">गर्भिणी+सु गो+सु (2.1.71)</option>
 <option value="gqRti+su go+su (2.1.65)">गृष्टि+सु गो+सु (2.1.65)</option>
 <option value="go+Am cawvAriMSaw+su (2.2.8)">गो+आम् चत्वारिंशत्+सु
 (2.2.8)</option>
 <option value="wallaja+su go+su (2.1.66)">तल्लज+सु गो+सु (2.1.66)</option>
 <option value="go+fas xoha+su (2.2.8)">गो+डस् दोह+सु (2.2.8)</option>
 <option value="go+fas paxa+su (2.2.8)">गो+डस् पद+सु (2.2.8)</option>
 <option value="prakANda+su go+su (2.1.66)">प्रकाण्ड+सु गो+सु

(2.1.66)</option>
 <option value="go+as balIvarxa+as (2.2.29)">गो+अस् बलीवर्द+अस्
 (2.2.29)</option>
 <option value="baRkayaNI+su go+su (2.1.65)">बष्कयणी+सु गो+सु
 (2.1.65)</option>
 <option value="go+su vqnxAraka+su (2.1.62)">गो+सु वृन्दारक+सु
 (2.1.62)</option>
 <option value="go+Am SAIA+su (2.2.8)">गो+आम् शाला+सु (2.2.8)</option>
 <option value="go+su aSva+su (2.2.29)">गो+सु अश्व+सु </option>
 <option value="goRTa+fi Svan+su (2.1.40)">गोष्ठ+ङि श्वन्+सु </option>
 <option value="go+Am paxa+su (2.2.8)">गो+आम् पद+सु (2.2.8)</option>
 <option value="go+fas paxa+su (2.2.8)">गो+ङस् पद+सु (2.2.8)</option>
 <option value="gOrI+fas pawI+su (2.2.8)">गौरी+ङस् पति+सु (2.2.8)</option>
 <option value="grAma+Am gamin+su (2.1.24)">ग्राम+आम् गमिन्+सु
 (2.1.24)</option>
 <option value="grAmaNI+su xqRti+su (2.2.24)">ग्रामणि+सु दृष्टि+सु
 (2.2.24)</option>
 <option value="grAmaNI+fas puwra+su (2.2.8)">ग्रामणी+ङस् पुत्र+सु
 (2.2.8)</option>
 <option value="grAma+fas wakRan+su (2.2.8)">ग्राम+ङस् तक्षन्+सु
 (2.2.8)</option>
 <option value="grAma+fi vAsin (2.2.19)">ग्राम+ङि वासिन् (2.2.19)</option>
 <option value="grAma+fas arXa+su (2.2.8)">ग्राम+ङस् अर्ध+सु (2.2.8)</option>
 <option value="grAma+fi vAsa+su (2.1.40)">ग्राम+ङि वास+सु (2.1.40)</option>

 <option value="grIRma+su vasanwa+su (2.2.29)">ग्रीष्म+सु वसन्त+सु
 (2.2.29)</option>
 <option value="Gana+su SyAma+su (2.1.55)">घन+सु श्याम+सु
 (2.1.55)</option>
 <option value="Gqwa+fas ganXa+su (2.2.24)">घृत+ङस् गन्ध+सु
 (2.2.24)</option>
 <option value="cakra+su niwamba+su (2.2.24)">चक्र+सु नितम्ब+सु
 (2.2.24)</option>
 <option value="cakra+fi baxXa+su (2.1.40)">चक्र+ङि बद्ध+सु (2.1.40)</option>
 <option value="cakra+fi banXa+su (2.1.41)">चक्र+ङि बन्ध+सु (2.1.41)</option>

 <option value="cakra+O sakWi+O (2.2.24)">चक्र+औ सक्थि+औ
 (2.2.24)</option>
 <option value="cawurafga+tA yoga+su (2.1.30)">चतुरङ्ग+टा योग+सु
 (2.1.30)</option>
 <option value="canxragupwa+fas saBA+su (2.2.8)">चन्द्रगुप्त+ङस् सभा+सु
 (2.2.8)</option>
 <option value="canxra+su muKa+su (2.2.24)">चन्द्र+सु मुख+सु

(2.2.24)</option>
 <option value="carama+su puruRa+su (2.1.58)">चरम+सु पुरुष+सु
 (2.1.58)</option>
 <option value="cANdAla+jas mqwapA+jas (2.2.29)">चाण्डाल+जस् मृतपा+जस्
 (2.2.29)</option>
 <option value="ciwrA+jas go+jas (2.2.24)">चित्रा+जस् गो+जस् (2.2.24)</option>

 <option value="jarawI+su go+su (2.1.57)">जरती+सु गो+सु (2.1.57)</option>
 <option value="ciwrA+su jaraxgavI+su (2.2.24)">चित्रा+सु जरद्गवी+सु
 (2.2.24)</option>
 <option value="cora+fasi Baya+su (2.1.37)">चोर+डसि भय+सु (2.1.37)</option>

 <option value="cOra+fas kula+su (2.2.8)">चौर+डस् कुल+सु (2.2.8)</option>
 <option value="Cawra+su upAnah+O (2.2.29)">छत्र+सु उपानह्+औ
 (2.2.29)</option>
 <option value="janus+tA anXa+su (2.1.30)">जनुस्+टा अन्ध+सु
 (2.1.30)</option>
 <option value="jAyA+su pawI+su (2.2.29)">जाया+सु पति+सु (2.2.29)</option>
 <option value="jarawI+su ciwrA+su (2.1.57)">जरती+सु चित्रा+सु
 (2.1.57)</option>
 <option value="jaracciwrA+su go+su (2.2.24)">जरच्चित्रा+सु गो+सु
 (2.2.24)</option>
 <option value="jaraw+su go+su (2.1.49)">जरत्+सु गो+सु (2.1.49)</option>
 <option value="jaraw+jas nEyAyika+jas (2.1.49)">जरत्+जस् नैयायिक+जस्
 (2.1.49)</option>
 <option value="jalaja+O akRi+O (2.2.24)">जलज+औ अक्षि+औ
 (2.2.24)</option>
 <option value="jala+fas saras+su (2.2.8)">जल+डस् सरस्+सु (2.2.8)</option>
 <option value="jAwi+fas vAcaka+su (2.2.8)">जाति+डस् वाचक+सु
 (2.2.8)</option>
 <option value="jAwa+su ukRan+su (2.1.57)">जात+सु उक्षन्+सु
 (2.1.57)</option>
 <option value="jAmbava+su SAlUkinI+su (2.2.29)">जाम्बव+सु शालूकिनी+सु
 (2.2.29)</option>
 <option value="jAyA+su pawI+su (2.2.29)">जाया+सु पति+सु (2.2.29)</option>
 <option value="kadAra+su jEmini+su (2.1.57)">कडार+सु जैमिनि+सु
 (2.1.57)</option>
 <option value="jyowis+Am swoma+su (2.2.8)">ज्योतिस्+आम् स्तोम+सु
 (2.2.8)</option>
 <option value="wakRan+jas ayaskAra+jas (2.2.29)">तक्षन्+जस् अयस्कार+जस्
 (2.2.29)</option>
 <option value="wax+fas upari (2.2.8)">तद्+डस् उपरि (2.2.8)</option>
 <option value="wax+fas muKa+su (2.2.8)">तद्+डस् मुख+सु (2.2.8)</option>

<option value="wapwa+su rahas+su (2.1.57)">तप्त+सु रहस्+सु (2.1.57)</option>
 <option value="warafga+Byas apawraswa+su (2.1.38)">तरङ्ग+भ्यस् अपत्रस्त+सु
 (2.1.38)</option>
 <option value="wigmA+jas ruc+jas (2.2.24)">तिग्मा+जस् रुच्+जस्
 (2.2.24)</option>
 <option value="wimi+am gila (2.2.19)">तिमि+अम् गिल (2.2.19)</option>
 <option value="wimi+Am gilagila+su (2.2.8)">तिमि+आम् गिलगिल+सु
 (2.2.8)</option>
 <option value="wiras kqwa+su (2.2.18)">तिरस् कृत+सु (2.2.18)</option>
 <option value="wiras+kq+wvA (2.2.18)">तिरस्+कृ+त्वा (2.2.18)</option>
 <option value="wiRya+su punarvasu+O (2.2.29)">तिष्य+सु पुनर्वसु+औ
 (2.2.29)</option>
 <option value="wIrWa+fi XvAfkRa+su (2.1.42)">तीर्थ+ङि ध्वाङ्ग+सु
 (2.1.42)</option>
 <option value="wufgA+su nAsikA+su (2.2.24)">तुङ्गा+सु नासिका+सु
 (2.2.24)</option>
 <option value="wulya+su mahaw+su (2.1.68)">तुल्य+सु महत्+सु
 (2.1.68)</option>
 <option value="wulya+su Svewa+su (2.1.68)">तुल्य+सु श्वेत+सु
 (2.1.68)</option>
 <option value="wqna+su jamBa+su (2.2.24)">तृन+सु जम्भ+सु (2.2.24)</option>
 <option value="wri+jas cawvAriMSaw+su (2.2.29)">त्रि+जस् चत्वारिंशत्+सु
 (2.2.29)</option>
 <option value="wri+jas xaSan+as (2.2.29)">त्रि+जस् दशन्+अस्
 (2.2.29)</option>
 <option value="wri+jas kakuxa+jas (2.2.24)">त्रि+जस् ककुद+जस्
 (2.2.24)</option>
 <option value="wriBuvana+fas viXAwq+su (2.2.8)">त्रिभुवन+ङस् विधातृ+सु
 (2.2.8)</option>
 <option value="wri+ijas mUrXan+jas (2.2.24)">त्रि+इजस् मूर्धन्+जस्
 (2.2.24)</option>
 <option value="wri+jas Sawa+su (2.2.29)">त्रि+जस् शत+सु (2.2.29)</option>
 <option value="wri+jas Sukla+jas (2.2.24)">त्रि+जस् शुक्ल+जस्
 (2.2.24)</option>
 <option value="wri+jas irAvawI+jas (2.2.24)">त्रि+जस् इरावती+जस्
 (2.2.24)</option>
 <option value="wri+jas aSIwi+su (2.2.29)">त्रि+जस् अशीति+सु
 (2.2.29)</option>
 <option value="wri+Am Ayus+su (2.2.8)">त्रि+आम् आयुस्+सु (2.2.8)</option>
 <option value="wvac+su sraj+su (2.2.29)">त्वच्+सु स्रज्+सु (2.2.29)</option>
 <option value="wvac+fi sAra+su (2.2.24)">त्वच्+ङि सार+सु (2.2.24)</option>

<option value="xakRina+fi Irma+su (2.2.24)">दक्षिन+डि ईर्म+सु
 (2.2.24)</option>
 <option value="pANi+fi xaNda+su (2.2.24)">पाणि+डि दण्ड+सु
 (2.2.24)</option>
 <option value="xANda+Bis xaNda+Bis (2.2.27)">दण्ड+भिस् दण्ड+भिस्
 (2.2.27)</option>
 <option value="xawwA+su BAryA+su (2.1.57)">दत्ता+सु भार्या+सु
 (2.1.57)</option>
 <option value="xawwA+su BAryA+su (2.2.24)">दत्ता+सु भार्या+सु
 (2.2.24)</option>
 <option value="xawwA+su yuvawi+su (2.2.29)">दत्ता+सु युवति+सु
 (2.2.29)</option>
 <option value="xaXi+fas Gata+su (2.2.8)">दधि+डस् घट+सु (2.2.8)</option>
 <option value="xaXi+su Gqwa+su (2.2.29)">दधि+सु घृत+सु (2.2.29)</option>
 <option value="xaXi+su payas+su (2.2.29)">दधि+सु पयस्+सु (2.2.29)</option>
 <option value="xaXi+tA oxana+su (2.1.34)">दधि+टा ओदन+सु
 (2.1.34)</option>
 <option value="xanwa+Am leKaka+su (2.2.17)">दन्त+आम् लेखक+सु
 (2.2.17)</option>
 <option value="xarBa+Am anUpa+su (2.2.8)">दर्भ+आम् अनूप+सु
 (2.2.8)</option>
 <option value="xarSanIyA+fas mAwq+su (2.2.8)">दर्शनीया+डस् मातृ+सु
 (2.2.8)</option>
 <option value="xarSanIyA+Am mAnin (2.2.19)">दर्शनीया+आम् मानिन्
 (2.2.19)</option>
 <option value="xaSan+jas KatvA+jas (2.2.24)">दशन्+जस् खट्वा+जस्
 (2.2.24)</option>
 <option value="xAkRi+Am kanWA+su (2.2.8)">दाक्षि+आम् कन्था+सु
 (2.2.8)</option>
 <option value="xAra+jas go+jas (2.2.29)">दार+जस् गो+जस् (2.2.29)</option>
 <option value="xAru+Am vaha+su (2.2.8)">दारु+आम् वह+सु (2.2.8)</option>
 <option value="xAsI+fas puwra+su (2.2.8)">दासी+डस् पुत्र+सु (2.2.8)</option>
 <option value="xiv+fi ja (2.2.19)">दिव्+डि ज (2.2.19)</option>
 <option value="xiv+fas xAsa+su (2.2.8)">दिव्+डस् दास+सु (2.2.8)</option>
 <option value="xiS+fas xaNda+su (2.2.8)">दिश्+डस् दण्ड+सु (2.2.8)</option>
 <option value="xIrGajA+su BAryA+su (2.2.24)">दीर्घजा+सु भार्या+सु
 (2.2.24)</option>
 <option value="xIrGa+O nayana+O (2.2.24)">दीर्घ+औ नयन+औ
 (2.2.24)</option>
 <option value="xIrGA+su rAwri+su (2.2.24)">दीर्घा+सु रात्रि+सु
 (2.2.24)</option>
 <option value="xIrGa+O sakWi+O (2.2.24)">दीर्घ+औ सक्थि+औ
 (2.2.24)</option>

<option value="xuKa+am awIwa+su (2.1.24)">दुख+अम् अतीत+सु
 (2.1.24)</option>
 <option value="xur brAhmana+su (2.2.18)">दुर् ब्राह्मन+सु (2.2.18)</option>
 <option value="xus hali+su (2.2.24)">दुस् हलि+सु (2.2.24)</option>
 <option value="xur hqxaya+su (2.2.24)">दुर् हृदय+सु (2.2.24)</option>
 <option value="xuhiwq+fas paw+su (2.2.8)">दुहितृ+डस् पति+सु
 (2.2.8)</option>
 <option value="nIla+su uwpala+su (2.1.57)">नील+सु उत्पल+सु
 (2.1.57)</option>
 <option value="na ekaXA (2.1.4)">न एकधा (2.1.4)</option>
 <option value="nyagroXa+Am vana+su (2.2.8)">न्यग्रोध+आम् वन+सु
 (2.2.8)</option>
 <option value="pAxa+Am Sabxa+su (2.2.8)">पाद+आम् शब्द+सु
 (2.2.8)</option>
 <option value="pAxa+ByAm ga (2.2.19)">पाद+भ्याम् ग (2.2.19)</option>
 <option value="pAxa+O aRTIvaw+O (2.2.29)">पाद+औ अष्टीवत्+औ
 (2.2.29)</option>
 <option value="pAxa+ByAm Aji (2.2.19)">पाद+भ्याम् आजि (2.2.19)</option>
 <option value="pAxa+ByAm upahawa+su (2.1.32)">पाद+भ्याम् उपहत+सु
 (2.1.32)</option>
 <option value="pAxa+ByAm ga (2.2.19)">पाद+भ्याम् ग (2.2.19)</option>
 <option value="pAxa+Am GoRa+su (2.2.8)">पाद+आम् घोष+सु (2.2.8)</option>

 <option value="pAxa+fas hima+su (2.2.8)">पाद+डस् हिम+सु (2.2.8)</option>
 <option value="paxma+su nABi+i (2.2.24)">पद्म+सु नाभि+इ (2.2.24)</option>
 <option value="pAxa+su niRka+su (2.1.57)">पाद+सु निष्क+सु (2.1.57)</option>

 <option value="pAxa+tA miSra+su (2.1.31)">पाद+टा मिश्र+सु (2.1.31)</option>
 <option value="parama+fi kAraka+i (2.1.61)">परम+ङि कारक+इ
 (2.1.61)</option>
 <option value="parama+su go+su (2.1.61)">परम+सु गो+सु (2.1.61)</option>
 <option value="paramA+su Xur+su (2.1.61)">परमा+सु धुर+सु (2.1.61)</option>

 <option value="parama+su puruRa+su (2.1.61)">परम+सु पुरुष+सु
 (2.1.61)</option>
 <option value="parama+su rAjan+su (2.1.61)">परम+सु राजन्+सु
 (2.1.61)</option>
 <option value="pAcikA+su swrI+su (2.1.57)">पाचिका+सु स्त्री+सु
 (2.1.57)</option>
 <option value="pANini+fas upajFA+su (2.2.8)">पाणिनि+डस् उपज्ञा+सु
 (2.2.8)</option>
 <option value="pANi+O pAxa+O (2.2.29)">पाणि+औ पाद+औ (2.2.29)</option>

<option value="pUrva+su vEyAkarana+su (2.1.58)">पूर्व+सु वैयाकरण+सु
 (2.1.58)</option>
 <option value="gafgA+fas maXya+su (2.1.18)">गङ्गा+डस् मध्य+सु
 (2.1.18)</option>
 <option value="maXya+fi guru+su (2.1.40)">मध्य+ङि गुरु+सु (2.1.40)</option>
 <option value="manas+tA gupwA+su (2.1.32)">मनस्+टा गुप्ता+सु
 (2.1.32)</option>
 <option value="manas+tA AjFAyin (2.2.19)">मनस्+टा आज्ञायिन्
 (2.2.19)</option>
 <option value="mahaw+fas sevA+su (2.2.8)">महत्+डस् सेवा+सु
 (2.2.8)</option>
 <option value="mahaw+su xeva+su (2.1.61)">महत्+सु देव+सु (2.1.61)</option>
 <option value="mahawI+su Xur+su (2.2.24)">महती+सु धुर्+सु (2.2.24)</option>

 <option value="mahawI+su navamI+su (2.1.61)">महती+सु नवमी+सु
 (2.1.61)</option>
 <option value="mahaw+su anas+su (2.1.61)">महत्+सु अनस्+सु
 (2.1.61)</option>
 <option value="mahaw+su paWin+su (2.1.61)">महत्+सु पथिन्+सु
 (2.1.61)</option>
 <option value="mahaw+su brahman+su (2.1.61)">महत्+सु ब्रह्मन्+सु
 (2.1.61)</option>
 <option value="mahaw+su yaSas+su (2.2.24)">महत्+सु यशस्+सु
 (2.2.24)</option>
 <option value="mahaw+su rAjan+su (2.1.61)">महत्+सु राजन्+सु
 (2.1.61)</option>
 <option value="mahaw+tA viSiRta+su (2.1.32)">महत्+टा विशिष्ट+सु
 (2.1.32)</option>
 <option value="saras+fi ja (2.2.19)">सरस्+ङि ज (2.2.19)</option>
 <option value="rAkRasI+Bis saha (2.2.28)">राक्षसी+भिस् सह (2.2.28)</option>
 <option value="sarvikA+su BAryA+su (2.2.24)">सर्विका+सु भार्या+सु
 (2.2.24)</option>
 <option value="su ganXa+jas (2.2.24)">सु गन्ध+जस् (2.2.24)</option>
 <option value="su xanwa+jas (2.2.24)">सु दन्त+जस् (2.2.24)</option>
 <option value="surARtra+sup brahman+su (2.1.40)">सुराष्ट्र+सुप् ब्रह्मन्+सु
 (2.1.40)</option>
 <option value="su vaXU+su (2.2.24)">सु वधू+सु (2.2.24)</option>
 <option value="su sanXi+su (2.2.18)">सु सन्धि+सु (2.2.18)</option>
 <option value="su sAman+su (2.2.24)">सु सामन्+सु (2.2.24)</option>
 <option value="su sakWi+O (2.2.24)">सु सक्थि+औ (2.2.24)</option>
 <option value="su sakWi+su (2.2.24)">सु सक्थि+सु (2.2.24)</option>
 <option value="su hali+su (2.2.24)">सु हलि+सु (2.2.24)</option>
 <option value="su hali+su (2.2.24)">सु हलि+सु (2.2.24)</option>

```

<option value="su hqxaya+su (2.2.24)">सु हृदय+सु (2.2.24)</option>
<option value="su hqxaya+su (2.2.24)">सु हृदय+सु (2.2.24)</option>
<option value="samAna+su uxara+su (2.1.58)">समान+सु उदर+सु
(2.1.58)</option>
<option value="snAwa+su anulipwa+su (2.1.49)">स्नात+सु अनुलिप्त+सु
(2.1.49)</option>
<option value="hari+su canxra+su (2.1.55)">हरि+सु चन्द्र+सु (2.1.55)</option>
<option value="hari+fas sakWi+su (2.2.8)">हरि+डस् सक्थि+सु (2.2.8)</option>
<option value="hari+su hara+su (2.2.29)">हरि+सु हर+सु (2.2.29)</option>
<option value="hala+fi wripaxikA+su (2.1.44)">हल+डि त्रिपदिका+सु
(2.1.44)</option>
<option value="haswa+fi banXa+su (2.1.41)">हस्त+डि बन्ध+सु
(2.1.41)</option>
<option value="haswin+fas varcas+su (2.2.8)">हस्तिन्+डस् वर्चस्+सु
(2.2.8)</option>
<option value="hqxaya+fas Soka+su (2.2.8)">हृदय+डस् शोक+सु
(2.2.8)</option>
<option value="hqxaya+fas roga+su (2.2.8)">हृदय+डस् रोग+सु (2.2.8)</option>
<option value="hqxaya+fas leKa+su (2.2.8)">हृदय+डस् लेख+सु
(2.2.8)</option>
<option value="hqxaya+fas lAsa+su (2.2.8)">हृदय+डस् लास+सु
(2.2.8)</option>
<option value="hqxaya+fas leKa (2.2.19)">हृदय+डस् लेख (2.2.19)</option>

</select>
</td>
</tr>
<tr>
<td>
<input type="hidden" name="encoding" id="encoding" value="DVNG">
<input type="submit" name="submit" value="submit" id="submit_query"
onclick="return samAsaprakAracgi()"/>
</td>
</tr>
</table>
</form>
</td>
<td>
<table border="1">
<tr>
<td> <div id="output1"> </div></td>
</tr>
<tr>

```

```

<td><div id="output2"> </div></td>
</tr>
<tr>
<td><div id="output3"> </div></td>
</tr>
<tr>
<td><div id="output4"> </div></td>
</tr>
<tr>
<td> <div id="output5"> </div></td>
</tr>
<tr>
<td> <div id="output6"> </div></td>
</tr>
<tr>
<td> <div id="output7"> </div></td>
</tr>
</table>
</center>
</div>
<center>
<p> <span class="cons">© 2012-15 Pavankumar Satuluri & <a
href="http://sanskrit.uohyd.ernet.in/faculty/amba">Amba Kulkarni</a>
</span>
<a href="http://localhost/htdocs/scl/index.html">Samsaadhanii</a> </p>
</center>
</body>
</html>

```

Callegiscript.js

```
function samAsaprakAracgi() {
  $("#output1").html("");
  $("#output2").html("");
  $("#output3").html("");
  $("#output4").html("");
  $("#output5").html("");
  $("#output6").html("");
  $("#output7").html("");
  $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/samAsaprakAra.cgi",{encoding:$
("#encoding").val(),vigraha:$("#vigraha_vAky").val()} ,function(data,status){
  $("#output1").html(data)
});
return false;
};
```

```
function samAsAnwayescgi() {
  $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/samAsAnwa.cgi",{encodingsamAsAnwa:$
("#encodingsamAsAnwa").val(),vigraphasamAsAnwa:$
("#vigraphasamAsAnwa").val(), p1samAsAnwa:$("#p1samAsAnwa").val(),
p2samAsAnwa:$("#p2samAsAnwa").val(), s1samAsAnwa:$
("#s1samAsAnwa").val(), s2samAsAnwa:$("#s2samAsAnwa").val(),
viXAYakasUwrasamAsAnwa:$("#viXAYakasUwrasamAsAnwa").val(),
samAsaprakArasamAsAnwa:$("#samAsaprakArasamAsAnwa").val(),
samAsAnwa:$("#samAsAnwa").val(), sUwrasamAsAnwa:$
("#sUwrasamAsAnwa").val(),anssamAsAnwa:'Yes'},function(data,status){
  $("#output2").html(data)
});
return false;
};
```

```
function samAsAnwanocgi() {
  $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/samAsAnwa.cgi",{encodingsamAsAnwa:$
("#encodingsamAsAnwa").val(),vigraphasamAsAnwa:$
("#vigraphasamAsAnwa").val(), p1samAsAnwa:$("#p1samAsAnwa").val(),
p2samAsAnwa:$("#p2samAsAnwa").val(), s1samAsAnwa:$
("#s1samAsAnwa").val(), s2samAsAnwa:$("#s2samAsAnwa").val(),
viXAYakasUwrasamAsAnwa:$("#viXAYakasUwrasamAsAnwa").val(),
samAsaprakArasamAsAnwa:$("#samAsaprakArasamAsAnwa").val(),
```

```

samAsAnwa:$("#samAsAnwa").val(), sUwrasamAsAnwa:$
("#sUwrasamAsAnwa").val(), anssamAsAnwa:'No'},function(data,status){
    $("#output2").html(data)
});
return false;
};

function samAsAnwaNullcgi() {
    $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/samAsAnwa.cgi",{encodingsamAsAnwa:$
("#encodingsamAsAnwa").val(),vighrasamAsAnwa:$
("#vighrasamAsAnwa").val(), p1samAsAnwa:$("#p1samAsAnwa").val(),
p2samAsAnwa:$("#p2samAsAnwa").val(), s1samAsAnwa:$
("#s1samAsAnwa").val(), s2samAsAnwa:$("#s2samAsAnwa").val(),
viXAYakasUwrasamAsAnwa:$("#viXAYakasUwrasamAsAnwa").val(),
samAsaprakArasamAsAnwa:$("#samAsaprakArasamAsAnwa").val(),
samAsAnwa:$("#samAsAnwa").val(), sUwrasamAsAnwa:$
("#sUwrasamAsAnwa").val(), anssamAsAnwa:'NULL'},function(data,status){
    $("#output2").html(data)
});
return false;
};

function pUrvanipAwayescgi() {
    $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/pUrvanipAwa.cgi",{encodingpUrvanipAwa:$
("#encodingpUrvanipAwa").val(),vighrapUrvanipAwa:$
("#vighrapUrvanipAwa").val(), p1pUrvanipAwa:$("#p1pUrvanipAwa").val(),
p2pUrvanipAwa:$("#p2pUrvanipAwa").val(), s1pUrvanipAwa:$
("#s1pUrvanipAwa").val(), s2pUrvanipAwa:$("#s2pUrvanipAwa").val(),
viXAYakasUwrapUrvanipAwa:$("#viXAYakasUwrapUrvanipAwa").val(),
samAsaprakArapUrvanipAwa:$("#samAsaprakArapUrvanipAwa").val(),
samAsAnwapUrvanipAwa:$("#samAsAnwapUrvanipAwa").val(),
positionpUrvanipAwa:$("#positionpUrvanipAwa").val(),sUwrapUrvanipAwa:$
("#sUwrapUrvanipAwa").val(),dividpUrvanipAwa:$
("#dividpUrvanipAwa").val(),anspUrvanipAwa:'Yes'},function(data,status){
    var x = $("#dividpUrvanipAwa").val();
    $(x).html(data)
});
return false;
};

function pUrvanipAwanocgi() {

```

```

$.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/pUrvanipAwa.cgi",{encodingpUrvanipAwa:$
("#encodingpUrvanipAwa").val(),vighapUrvanipAwa:$
("#vighapUrvanipAwa").val(), p1pUrvanipAwa:$("#p1pUrvanipAwa").val(),
p2pUrvanipAwa:$("#p2pUrvanipAwa").val(), s1pUrvanipAwa:$
("#s1pUrvanipAwa").val(), s2pUrvanipAwa:$("#s2pUrvanipAwa").val(),
viXayakasUwrapUrvanipAwa:$("#viXayakasUwrapUrvanipAwa").val(),
samAsaprakArapUrvanipAwa:$("#samAsaprakArapUrvanipAwa").val(),
samAsAnwapUrvanipAwa:$("#samAsAnwapUrvanipAwa").val(),
positionpUrvanipAwa:$("#positionpUrvanipAwa").val(),sUwrapUrvanipAwa:$
("#sUwrapUrvanipAwa").val(),dividpUrvanipAwa:$
("#dividpUrvanipAwa").val(),anspUrvanipAwa:'No'},function(data,status){
    // var x = $("#sUwra").val();
    // alert(x);
    var x = $("#dividpUrvanipAwa").val();
    $(x).html(data)
});
return false;
};

```

```

function pUrvanipAwaNullcgi() {
$.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/pUrvanipAwa.cgi",{encodingpUrvanipAwa:$
("#encodingpUrvanipAwa").val(),vighapUrvanipAwa:$
("#vighapUrvanipAwa").val(), p1pUrvanipAwa:$("#p1pUrvanipAwa").val(),
p2pUrvanipAwa:$("#p2pUrvanipAwa").val(), s1pUrvanipAwa:$
("#s1pUrvanipAwa").val(), s2pUrvanipAwa:$("#s2pUrvanipAwa").val(),
viXayakasUwrapUrvanipAwa:$("#viXayakasUwrapUrvanipAwa").val(),
samAsaprakArapUrvanipAwa:$("#samAsaprakArapUrvanipAwa").val(),
samAsAnwapUrvanipAwa:$("#samAsAnwapUrvanipAwa").val(),
positionpUrvanipAwa:$("#positionpUrvanipAwa").val(),sUwrapUrvanipAwa:$
("#sUwrapUrvanipAwa").val(),dividpUrvanipAwa:$
("#dividpUrvanipAwa").val(),anspUrvanipAwa:'NULL'},function(data,status){
    // var x = $("#sUwra").val();
    // alert(x);
    var x = $("#dividpUrvanipAwa").val();
    $(x).html(data)
});
return false;
};
//

```

```

function upasarjanayescgi() {
$.post("http://localhost/cgi-

```

```

bin/scl/skt_gen/new_compounds/upasarjana.cgi", {encodingupasarjana:$
("#encodingupasarjana").val(),vighraupasarjana:$("#vighraupasarjana").val(),
p1upasarjana:$("#p1upasarjana").val(), p2upasarjana:$("#p2upasarjana").val(),
s1upasarjana:$("#s1upasarjana").val(), s2upasarjana:$("#s2upasarjana").val(),
viXAyakasUwraupasarjana:$("#viXAyakasUwraupasarjana").val(),
samAsaprakAraupasarjana:$("#samAsaprakAraupasarjana").val(),
samAsAnwaupasarjana:$("#samAsAnwaupasarjana").val(), positionupasarjana:
$("#positionupasarjana").val(),dividupasarjana:$
("#dividupasarjana").val(),ansupasarjana:'Yes'},function(data,status){
    var x = $("#dividupasarjana").val();
    $(x).html(data)
});
return false;
};

```

```

function upasarjananocgi() {
    $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/upasarjana.cgi", {encodingupasarjana:$
("#encodingupasarjana").val(),vighraupasarjana:$("#vighraupasarjana").val(),
p1upasarjana:$("#p1upasarjana").val(), p2upasarjana:$("#p2upasarjana").val(),
s1upasarjana:$("#s1upasarjana").val(), s2upasarjana:$("#s2upasarjana").val(),
viXAyakasUwraupasarjana:$("#viXAyakasUwraupasarjana").val(),
samAsaprakAraupasarjana:$("#samAsaprakAraupasarjana").val(),
samAsAnwaupasarjana:$("#samAsAnwaupasarjana").val(), positionupasarjana:
$("#positionupasarjana").val(),dividupasarjana:$
("#dividupasarjana").val(),ansupasarjana:'No'},function(data,status){
    var x = $("#dividupasarjana").val();
    $(x).html(data)
});
return false;
};

```

```

function upasarjanaNullcgi() {
    $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/upasarjana.cgi", {encodingupasarjana:$
("#encodingupasarjana").val(),vighraupasarjana:$("#vighraupasarjana").val(),
p1upasarjana:$("#p1upasarjana").val(), p2upasarjana:$("#p2upasarjana").val(),
s1upasarjana:$("#s1upasarjana").val(), s2upasarjana:$("#s2upasarjana").val(),
viXAyakasUwraupasarjana:$("#viXAyakasUwraupasarjana").val(),
samAsaprakAraupasarjana:$("#samAsaprakAraupasarjana").val(),
samAsAnwaupasarjana:$("#samAsAnwaupasarjana").val(), positionupasarjana:
$("#positionupasarjana").val(),dividupasarjana:$
("#dividupasarjana").val(),ansupasarjana:'NULL'},function(data,status){

```

```

// var x = $("#samAsAnwaupasarjana").val();
// alert(x);
var x = $("#dividupasarjana").val();
$(x).html(data)
});
return false;
};

function gawiyescgi() {
$.post("http://localhost/cgi-bin/scl/skt_gen/new_compounds/gawi.cgi",
{encodinggawi:$("#encodinggawi").val(),vigrahagawi:$("#vigrahagawi").val(),
p1gawi:$("#p1gawi").val(), p2gawi:$("#p2gawi").val(), s1gawi:$
("#s1gawi").val(), s2gawi:$("#s2gawi").val(), viXAyakasUwragawi:$
("#viXAyakasUwragawi").val(), samAsaprakAragawi:$
("#samAsaprakAragawi").val(), samAsAnwagawi:$
("#samAsAnwagawi").val(), positiongawi:$("#positiongawi").val(),gawisUwra:
$("#gawisUwra").val(),ansgawi:'Yes'},function(data,status){
    $("#output3").html(data)
});
return false;
};

function gawinocgi() {
$.post("http://localhost/cgi-bin/scl/skt_gen/new_compounds/gawi.cgi",
{encodinggawi:$("#encodinggawi").val(),vigrahagawi:$("#vigrahagawi").val(),
p1gawi:$("#p1gawi").val(), p2gawi:$("#p2gawi").val(), s1gawi:$
("#s1gawi").val(), s2gawi:$("#s2gawi").val(), viXAyakasUwragawi:$
("#viXAyakasUwragawi").val(), samAsaprakAragawi:$
("#samAsaprakAragawi").val(), samAsAnwagawi:$
("#samAsAnwagawi").val(), positiongawi:$("#positiongawi").val(),gawisUwra:
$("#gawisUwra").val(),ansgawi:'No'},function(data,status){
    $("#output3").html(data)
});
return false;
};

function gawiNullcgi() {
$.post("http://localhost/cgi-bin/scl/skt_gen/new_compounds/gawi.cgi",
{encodinggawi:$("#encodinggawi").val(),vigrahagawi:$("#vigrahagawi").val(),
p1gawi:$("#p1gawi").val(), p2gawi:$("#p2gawi").val(), s1gawi:$
("#s1gawi").val(), s2gawi:$("#s2gawi").val(), viXAyakasUwragawi:$
("#viXAyakasUwragawi").val(), samAsaprakAragawi:$
("#samAsaprakAragawi").val(), samAsAnwagawi:$

```

```

("#samAsAnwagawi").val(), positiongawi:$("#positiongawi").val(),gawisUwra:
$("#gawisUwra").val(),ansgawi:'NULL'},function(data,status){
    $("#output3").html(data)
});
return false;
};

```

```

function alukyescgi() {
    $.post("http://localhost/cgi-bin/scl/skt_gen/new_compounds/aluk.cgi",
    {encodingaluk:$("#encodingaluk").val(),p1aluk:$("#p1aluk").val(), p2aluk:$
    ($("#p2aluk").val(), s1aluk:$("#s1aluk").val(), s2aluk:$("#s2aluk").val(),
    samAsaprakAraaluk:$("#samAsaprakAraaluk").val(), samAsAnwaaluk:$
    ($("#samAsAnwaaluk").val(), alukviBARA:$("#alukviBARA").val(),sUwraaluk:
    ($("#sUwraaluk").val(),dividaluk:$
    ($("#dividaluk").val(),ansaluk:'Yes'},function(data,status){
        var x = ($("#dividaluk").val());
        $(x).html(data)
    });
    return false;
};

```

```

function aluknocgi() {
    $.post("http://localhost/cgi-bin/scl/skt_gen/new_compounds/aluk.cgi",
    {encodingaluk:$("#encodingaluk").val(),p1aluk:$("#p1aluk").val(), p2aluk:$
    ($("#p2aluk").val(), s1aluk:$("#s1aluk").val(), s2aluk:$("#s2aluk").val(),
    samAsaprakAraaluk:$("#samAsaprakAraaluk").val(), samAsAnwaaluk:$
    ($("#samAsAnwaaluk").val(), alukviBARA:$("#alukviBARA").val(),sUwraaluk:
    ($("#sUwraaluk").val(),dividaluk:$
    ($("#dividaluk").val(),ansaluk:'No'},function(data,status){
        var x = ($("#dividaluk").val());
        $(x).html(data)
    });
    return false;
};

```

```

function alukNullcgi() {
    $.post("http://localhost/cgi-bin/scl/skt_gen/new_compounds/aluk.cgi",
    {encodingaluk:$("#encodingaluk").val(),p1aluk:$("#p1aluk").val(), p2aluk:$
    ($("#p2aluk").val(), s1aluk:$("#s1aluk").val(), s2aluk:$("#s2aluk").val(),
    samAsaprakAraaluk:$("#samAsaprakAraaluk").val(), samAsAnwaaluk:$
    ($("#samAsAnwaaluk").val(), alukviBARA:$("#alukviBARA").val(),sUwraaluk:
    ($("#sUwraaluk").val(),dividaluk:$
    ($("#dividaluk").val(),ansaluk:'NULL'},function(data,status){

```

```

    var x = $("#dividaluk").val();
    $(x).html(data)
  });
  return false;
}

function pUrvapaxayescgi() {
  $.post("http://localhost/cgi-bin/scl/skt_gen/new_compounds/pUrvapaxa.cgi",{encodingpUrva:$("#encodingpUrva").val(),p1pUrva:$("#p1pUrva").val(), p2pUrva:$("#p2pUrva").val(), samAsaprakArapUrva:$("#samAsaprakArapUrva").val(), samAsAnwapUrva:$("#samAsAnwapUrva").val(), sUwrastrpUrva:$("#sUwrastrpUrva").val(),strmodpUrva:$("#strmodpUrva").val(),strunmodpUrva:$("#strunmodpUrva").val(),dividpUrva:$("#dividpUrva").val(),anspUrva:'Yes'},function(data,status){
    var x = $("#dividpUrva").val();
    $(x).html(data)
  });
  return false;
};

function pUrvapaxanocgi() {
  $.post("http://localhost/cgi-bin/scl/skt_gen/new_compounds/pUrvapaxa.cgi",{encodingpUrva:$("#encodingpUrva").val(),p1pUrva:$("#p1pUrva").val(), p2pUrva:$("#p2pUrva").val(), samAsaprakArapUrva:$("#samAsaprakArapUrva").val(), samAsAnwapUrva:$("#samAsAnwapUrva").val(), sUwrastrpUrva:$("#sUwrastrpUrva").val(),strmodpUrva:$("#strmodpUrva").val(),strunmodpUrva:$("#strunmodpUrva").val(),dividpUrva:$("#dividpUrva").val(),anspUrva:'No'},function(data,status){
    var x = $("#dividpUrva").val();
    $(x).html(data)
  });
  return false;
};

function pUrvapaxaNullcgi() {
  // alert("In pUrvapaxaNullcgi");
  // var x = $("#samAsAnwapUrva").val();
  // alert(x);
  $.post("http://localhost/cgi-bin/scl/skt_gen/new_compounds/pUrvapaxa.cgi",{encodingpUrva:

```

```

("#encodingpUrva").val(),p1pUrva:$("#p1pUrva").val(), p2pUrva:$
("#p2pUrva").val(), samAsaprakArapUrva:$("#samAsaprakArapUrva").val(),
samAsAnwapUrva:$("#samAsAnwapUrva").val(), sUwrastrpUrva:$
("#sUwrastrpUrva").val(),strmodpUrva:$
("#strmodpUrva").val(),strunmodpUrva:$
("#strunmodpUrva").val(),dividpUrva:$
("#dividpUrva").val(),anspUrva:'NULL'},function(data,status){
    var x = ($("#dividpUrva").val());
    $(x).html(data)
});
return false;
}

```

```

function uwwarapaxayescgi() {
    $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/uwwarapaxa.cgi",{encodinguwwara:$
("#encodinguwwara").val(),p1uwwara:$("#p1uwwara").val(), p2uwwara:$
("#p2uwwara").val(), samAsAnwauwwara:$
("#samAsAnwauwwara").val(),samAsaprakArauwwara:$
("#samAsaprakArauwwara").val(), sUwrastruwwara:$
("#sUwrastruwwara").val(),uwwarapaxastr:$
("#uwwarapaxastr").val(),dividuwwara:$
("#dividuwwara").val(),ansuwwara:'Yes'},function(data,status){
    var x = ($("#dividuwwara").val());
    $(x).html(data)
});
return false;
};

```

```

function uwwarapaxanocgi() {
    $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/uwwarapaxa.cgi",{encodinguwwara:$
("#encodinguwwara").val(),p1uwwara:$("#p1uwwara").val(), p2uwwara:$
("#p2uwwara").val(), samAsAnwauwwara:$
("#samAsAnwauwwara").val(),samAsaprakArauwwara:$
("#samAsaprakArauwwara").val(), sUwrastruwwara:$
("#sUwrastruwwara").val(),uwwarapaxastr:$
("#uwwarapaxastr").val(),dividuwwara:$
("#dividuwwara").val(),ansuwwara:'No'},function(data,status){
    var x = ($("#dividuwwara").val());
    $(x).html(data)
});
return false;
}

```

```

};
function uwwarapaxaNullcgi() {
    $.post("http://localhost/cgi-
bin/scl/skt_gen/new_compounds/uwwarapaxa.cgi",{encodinguwwara:$
("#encodinguwwara").val(),p1uwwara:$("#p1uwwara").val(), p2uwwara:$
("#p2uwwara").val(), samAsAnwauwwara:$
("#samAsAnwauwwara").val(),samAsaprakArauwwara:$
("#samAsaprakArauwwara").val(), sUwrastruwwara:$
("#sUwrastruwwara").val(),uwwarapaxastr:$
("#uwwarapaxastr").val(),dividuwwara:$
("#dividuwwara").val(),ansuwwara:'NULL'},function(data,status){
    var x = $("#dividuwwara").val();
    $(x).html(data)
});
return false;
}

```

Compound.css

```
.head_div{background:none repeat scroll 0px 0px #5678AA; color:#fff;}
h3{margin-top:1px !important; font-size:20px;}
.head_div p{font-size:16px !important; }
.text div{padding:5px;}
input[type="submit"]{margin-top:5px;}
select{margin-top:10px; width:300px !important; padding:3px !important;}
.listdiv{margin-top:5px; height: 500px}
#output3, #output4, #output5, #output7, #output8, #output6{
    font-size:15px;
    width:800px;
    height:90px;
}
#output1{
    font-size:15px;
    width:800px;
    height:70px;
}
#output2{
    font-size:15px;
    width:800px;
    height:110px;
}
```

Samasaprakara.cgi

```
#!/usr/bin/perl

my $myPATH = "/home/pavan/scl/skt_gen/new_compounds";

use warnings;
use CGI ':standard';

my $cgi = new CGI;

print $cgi->header(-type => 'text/html',
                  -charset => 'utf-8');

if (param) {
    my $encoding = param("encoding");
    my $avighraha = param("vighraha");

    $avighraha =~ /^(([\^+ ]+)\+?([\^ ]*) ([\^+ ]+)\+?([\^ ]*) \((([1-8].[1-4].[0-9]+\))\))/;
    my $p1 = $1;
    my $s1 = $2;
    my $p2 = $3;
    my $s2 = $4;
    my $sUwra = $5;

    $avighraha =~ s/ \((([1-8].[1-4].[0-9]+\))\)//;

    #print "Calling samAsaprakAra.out";
    my $cmd =
"$myPATH/samAsaprakAra.out \"$encoding\" \"$avighraha\" \"$p1\" \"$s1\" \"$s2\" \"$sUwra\"";
    system($cmd);
}
```

Samaasanta.cgi

```
#!/usr/bin/perl

my $myPATH = "/home/pavan/scl/skt_gen/new_compounds/";

require "$myPATH/cnvrt2utf.pl";
require "$myPATH/cnvrt2utfr.pl";

use warnings;
use CGI ':standard';

my $cgi = new CGI;

print $cgi->header(-type => 'text/html',
                  -charset => 'utf-8');

if (param) {
    my $encodingsamAsAnwa = param("encodingsamAsAnwa");
    my $avighrasamAsAnwa = param("vighrasamAsAnwa");
    my $p1samAsAnwa = param("p1samAsAnwa");
    my $p2samAsAnwa = param("p2samAsAnwa");
    my $s1samAsAnwa = param("s1samAsAnwa");
    my $s2samAsAnwa = param("s2samAsAnwa");
    my $viXAyakasUwrasamAsAnwa = param("viXAyakasUwrasamAsAnwa");
    my $samAsaparakArasamAsAnwa = param("samAsaparakArasamAsAnwa");
    my $samAsAnwa = param("samAsAnwa");
    my $sUwrasamAsAnwa = param("sUwrasamAsAnwa");
    my $anssamAsAnwa = param("anssamAsAnwa");

    if ($anssamAsAnwa eq "No") {$samAsAnwa = "--"; $sUwrasamAsAnwa =
"";}

    my $avighraha_utf = &cnvrt2utf($avighrasamAsAnwa);
    my $samAsAnwa_utf = &cnvrt2utf($samAsAnwa);

    if ($encodingsamAsAnwa eq "RMN") {
        my $avighraha_utfr = &cnvrt2utfr($avighraha_utf);
        my $samAsAnwa_utfr = &cnvrt2utfr($samAsAnwa);
        my $sUwra_utfr = &cnvrt2utfr($sUwrasamAsAnwa);
    }

    print "<table>";
```

```

print "<tr>";

if ($encodingsamAsAnwa eq "RMN") {
    print "<td> <font color=\"blue\">$avighraha_utfr</font></td>";
} else {
    print "<td> <font color=\"blue\">$avighraha_utf</font></td>";
}

if($samAsAnwa eq "") {
    if ($encodingsamAsAnwa eq "RMN") {
        print "<td><font color=\"green\"> samāsāntapratyayah --
</font></td>";
    } else {
        print "<td><font color=\"green\">समासान्तप्रत्ययः -- </font></td>";
    }
} else {
    if ($encodingsamAsAnwa eq "RMN") {
        print "<td><font color=\"green\"> samāsāntapratyayah
$samAsAnwa_utfr </font></td>";
    } else {
        print "<td><font color=\"green\">समासान्तप्रत्ययः $samAsAnwa_utf
</font></td>";
    }
}
}
if ($encodingsamAsAnwa eq "RMN") {
    print "<td><font color=\"red\">$sUwra_utfr </font></td>";
} else {
    print "<td><font color=\"red\">$sUwrasamAsAnwa </font></td>";
}
print "</tr>";
print "<tr>";
if ($encodingsamAsAnwa eq "RMN") {
    print "<td> <font color=\"blue\">$avighraha_utfr ";
} else {
    print "<td> <font color=\"blue\">$avighraha_utf ";
}

if($samAsAnwa ne "") {
    if ($encodingsamAsAnwa eq "RMN") {
        print " $samAsAnwa_utfr </font></td>";
    } else {
        print " $samAsAnwa_utf </font></td>";
    }
}

```

```

} else { print "</font></td>";}

if ($encodingsamAsAnwa eq "RMN"){
  print "<td><font color=\"green\"> prātipadikasaṃjñā </font></td>";
  print "<td><font color=\"red\"> kṛttaddhitasamāsāśca 1.2.46
</font></td>";
} else {
  print "<td><font color=\"green\"> प्रातिपदिकसंज्ञा</font></td>";
  print "<td><font color=\"red\"> कृत्तद्धितसमासाश्च 1.2.46</font></td>";
}
my $cmd =
"$myPATH/upasarjana.out \"$encodingsamAsAnwa\" \"$avigrahasamAsAnwa\
\" \"$p1samAsAnwa\" \"$s1samAsAnwa\" \"$p2samAsAnwa\" \"$s2samAsAnw
a\" \"$viXayakasUwrasamAsAnwa\" \"$samAsaparakArasamAsAnwa\" \"$sam
AsAnwa\"";
system($cmd);
}

```

Upasarjana.cgi

```
#!/usr/bin/perl

my $myPATH = "/home/pavan/scl/skt_gen/new_compounds";
require "$myPATH/cnvrt2utf.pl";
require "$myPATH/cnvrt2utfr.pl";

use warnings;
use CGI ':standard';

my $cgi = new CGI;

print $cgi->header(-type => 'text/html',
                  -charset => 'utf-8');

if (param) {
    my $encodingupasarjana = param("encodingupasarjana");
    my $avighraupasarjana = param("avighraupasarjana");
    my $p1upasarjana = param("p1upasarjana");
    my $p2upasarjana = param("p2upasarjana");
    my $s1upasarjana = param("s1upasarjana");
    my $s2upasarjana = param("s2upasarjana");
    my $samAsaprakAraupasarjana = param("samAsaprakAraupasarjana");
    my $samAsAnwaupasarjana = param("samAsAnwaupasarjana");
    my $viXayakasUwraupasarjana = param("viXayakasUwraupasarjana"); #
    Not needed
    my $positionupasarjana = param("positionupasarjana");
    my $dividupasarjana = param("dividupasarjana");
    my $ansupasarjana = param("ansupasarjana");

    #print "in upasarjana.cgi samAsAnwa = $samAsAnwaupasarjana<br />";
    my $p1utf = &cnvrt2utf($p1upasarjana);
    my $p2utf = &cnvrt2utf($p2upasarjana);
    my $avighra2utf = &cnvrt2utf($avighraupasarjana);

    if ($encodingupasarjana eq "RMN") {
        my $p1utfr = &cnvrt2utfr($p1utf);
        my $p2utfr = &cnvrt2utfr($p2utf);
        my $avighra2utfr = &cnvrt2utfr($avighra2utf);
    }

    $dividupasarjana =~ s/^#output//;
```

```

$dividupasarjana++;

print "<table>";
if ($ansupasarjana eq "No") {$positionupasarjana = 3 - $positionupasarjana;}
# 1 -> 2; 2 -> 1
if ($positionupasarjana == 3) { $positionupasarjana = 0;}

if ($encodingupasarjana eq "RMN") {
    print "<tr><td><font color=\"blue\">$avighraha2utf</font></td>";
} else {
    print "<tr><td><font color=\"blue\">$avighraha2utf</font></td>";
}
if ($positionupasarjana == 1) {
    if ($encodingupasarjana eq "RMN") {
        print "<td><font color=\"green\"> upasarjanasaṃjñā : $p1utf
<font></td>";
    } else {
        print "<td><font color=\"green\"> उपसर्जनसञ्ज्ञा : $p1utf <font></td>";
    }
}
elseif ($positionupasarjana == 2) {
    if ($encodingupasarjana eq "RMN") {
        print "<td><font color=\"green\"> upasarjanasaṃjñā : $p2utf
<font></td>";
    } else {
        print "<td><font color=\"green\"> उपसर्जनसञ्ज्ञा : $p2utf <font></td>";
    }
}
# Vighraha vAkyā with changed word order
$avighrahaupasarjana = $p2upasarjana;
if ($s2upasarjana ne "") {$avighrahaupasarjana .= "+".$s2upasarjana;}
$avighrahaupasarjana .= " ".$p1upasarjana;
if ($s1upasarjana ne "") {$avighrahaupasarjana .= "+".$s1upasarjana;}
$avighraha2utf = &cnvrt2utf($avighrahaupasarjana);

if ($encodingupasarjana eq "RMN") {
    $avighraha2utf = &cnvrt2utf($avighraha2utf);
}

$tmp = $p1upasarjana;
$p1upasarjana = $p2upasarjana;
$p2upasarjana = $tmp;

$tmp = $s1upasarjana;

```

```

    $s1upasarjana = $s2upasarjana;
    $s2upasarjana = $tmp;
}
elseif ($positionupasarjana eq "0") {
    print "Can not decide the upasarjana saFjFA. <br>";
    print "Compound can not be formed.<br>";
    print "Exiting ...";
    exit;
}
if ($encodingupasarjana eq "RMN") {
    print "<td> <font color=\"red\"> prathamānirdiṣṭaṃ samāsa
upasarjanam1.2.43 </td></font>";
} else {
    print "<td> <font color=\"red\"> प्रथमानिर्दिष्टं समास उपसर्जनम् 1.2.43
</td></font>";
}
print "</tr>";
if ($encodingupasarjana eq "RMN") {
    print "<tr><td><font color=\"blue\">$avighraha2utf</font></td>";
} else {
    print "<tr><td><font color=\"blue\">$avighraha2utf</font></td>";
}
if ($encodingupasarjana eq "RMN"){
    print "<td><font color=\"green\"> pūrvanipātaḥ </font></td>";
} else {
    print "<td><font color=\"green\"> पूर्वनिपातः</font></td>";
}
if ($encodingupasarjana eq "RMN") {
    print "<td><font color=\"red\"> upasarjanaṃ pūrvam 2.2.30</font></td>";
} else {
    print "<td><font color=\"red\"> उपसर्जनं पूर्वम् 2.2.30</font></td>";
}

#print "calling aluk with samAsAnwa = $samAsAnwaupasarjana<br/>";
# Note the changed word order of components
my $cmd =
"$myPATH/aluk.out \"$encodingupasarjana\" \"$avighrahaupasarjana\" \"$p1upa
sarjana\" \"$s1upasarjana\" \"$p2upasarjana\" \"$s2upasarjana\" \"$samAsaprak
Araupasarjana\" \"$samAsAnwaupasarjana\" $dividupasarjana";
system($cmd);
}

```

Gati.cgi

```
#!/usr/bin/perl

$myPATH="/home/pavan/scl/skt_gen/new_compounds";
require "$myPATH/cnvrt2utf.pl";
require "$myPATH/cnvrt2utfr.pl";

use warnings;
use CGI ':standard';

my $cgi = new CGI;

print $cgi->header(-type => 'text/html',
                  -charset => 'utf-8');

if (param) {
    my $encodinggawi = param("encodinggawi");
    my $avighragawi = param("vighragawi");
    my $p1gawi = param("p1gawi");
    my $p2gawi = param("p2gawi");
    my $s1gawi = param("s1gawi");
    my $s2gawi = param("s2gawi");
    my $samAsaprakAragawi = param("samAsaprakAragawi");
    my $samAsAnwagawi = param("samAsAnwagawi");
    my $viXAYakasUwragawi = param("viXAYakasUwragawi");
    my $positiongawi = param("positiongawi");
    my $gawisUwragawi = param("gawisUwra");
    my $ansgawi = param("ansgawi");

    my $p1utf = &cnvrt2utf($p1gawi);
    my $p2utf = &cnvrt2utf($p2gawi);
    my $avighraha2utf = &cnvrt2utf($avighragawi);

    if ($encodinggawi eq "RMN") {
        my $p1utfr = &cnvrt2utfr($p1utf);
        my $p2utfr = &cnvrt2utfr($p2utf);
        my $avighraha2utfr = &cnvrt2utfr($avighraha2utf);
        my $gawisUwrautfr = &cnvrt2utfr($gawisUwra);
    }

    print "<table>\n";
    print "<tr>";
```

```

if ($encodinggawi eq "RMN") {
    print "<td> <font color=\"blue\">$avighraha2utfr </font></td>\n";
} else {
    print "<td> <font color=\"blue\">$avighraha2utf </font></td>\n";
}

if ($ansgawi eq "No") {$positiongawi = 3 - $positiongawi;} # 1 -> 2; 2 -> 1
if ($positiongawi == 3) { $positiongawi = 0;}

if ($positiongawi == 1) {
    if ($encodinggawi eq "DVNG") {
        print "<td><font color=\"green\">गति संज्ञा : $p1utf </font></td>";
        print "<td> <font color=\"red\">$gawisUwra </font></td>\n";
    } else {
        print "<td><font color=\"green\"> gati saṃjñā : $p1utfr </font></td>";
        print "<td> <font color=\"red\">$gawisUwrautfr </font></td>\n";
    }
} elseif ($positiongawi == 2) {
    if ($encodinggawi eq "DVNG") {
        print "<td><font color=\"green\">गति संज्ञा : $p2utf </font></td>";
        print "<td> <font color=\"red\">$gawisUwra </font></td>\n";
    } else {
        print "<td><font color=\"green\"> gati saṃjñā : $p2utfr </font></td>";
        print "<td> <font color=\"red\">$gawisUwrautfr </font></td>\n";
    }
}
# vighraha vAkyā with changed order
$avighrahagawi = $p2gawi;
if ($s2gawi ne "") {$avighrahagawi .= "+".$s2gawi;}
$avighrahagawi .= " ".$p1gawi;
if ($s1gawi ne "") {$avighrahagawi .= "+".$s1gawi;}
} elseif ($positiongawi == 0) {
    print "Can not decide the upasarjana saFjFA. <br>";
    print "Compound can not be formed.<br>";
    print "Exiting ...";
    exit;
}

print "<td><form name=\"f3\" method=\"post\" />\n
<input type=\"hidden\" name=\"encodingupasarjana\"
id=\"encodingupasarjana\" value=\"$encodinggawi\" />\n
<input type=\"hidden\" name=\"vighraupasarjana\"
id=\"vighraupasarjana\" value=\"$avighrahagawi\" />\n
<input type=\"hidden\" name=\"p1upasarjana\" id=\"p1upasarjana\"

```

```

value="\$p1gawi\" />\n
    <input type=\"hidden\" name=\"p2upasarjana\" id=\"p2upasarjana\"
value="\$p2gawi\" />\n
    <input type=\"hidden\" name=\"s1upasarjana\" id=\"s1upasarjana\"
value="\$s1gawi\" />\n
    <input type=\"hidden\" name=\"s2upasarjana\" id=\"s2upasarjana\"
value="\$s2gawi\" />\n
    <input type=\"hidden\" name=\"samAsaparakAraupasarjana\"
id=\"samAsaparakAraupasarjana\" value=\"$samAsaparakAragawi\" />\n
    <input type=\"hidden\" name=\"samAsAnwaupasarjana\"
id=\"samAsAnwaupasarjana\" value=\"$samAsAnwagawi\" />\n
    <input type=\"hidden\" name=\"viXAyakasUwraupasarjana\"
id=\"viXAyakasUwraupasarjana\" value=\"$viXAyakasUwragawi\" />\n
    <input type=\"hidden\" name=\"positionupasarjana\"
id=\"positionupasarjana\" value=\"$positiongawi\" />\n
    <input type=\"hidden\" name=\"dividupasarjana\" id=\"dividupasarjana\"
value=\"#output4\" />\n";
    if ($encodinggawi eq "DVNG") {
        print "<input type=\"submit\" name=\"ansupasarjana\" value=\"उपसर्जन
संज्ञा निर्धारण\" id=\"ansupasarjana\" onclick=\"return upasarjanaNullcgi()\"
/>\n";
    } else {
        print "<input type=\"submit\" name=\"ansupasarjana\"
value=\"upasarjana samjñā nirdhāraṇa\" id=\"ansupasarjana\" onclick=\"return
upasarjanaNullcgi()\" />\n";
    }
    print "</form></td></tr></table>";
}

```

Purvanipata.cgi

```
#!/usr/bin/perl

my $myPATH = "/home/pavan/scl/skt_gen/new_compounds";
require "$myPATH/cnvrt2utf.pl";
require "$myPATH/cnvrt2utfr.pl";

use warnings;
use CGI ':standard';

my $cgi = new CGI;

print $cgi->header(-type => 'text/html',
                  -charset => 'utf-8');

if (param) {
    my $encodingpUrvanipAwa = param("encodingpUrvanipAwa");
    my $avighrahapUrvanipAwa = param("avighrahapUrvanipAwa");
    my $p1pUrvanipAwa = param("p1pUrvanipAwa");
    my $p2pUrvanipAwa = param("p2pUrvanipAwa");
    my $s1pUrvanipAwa = param("s1pUrvanipAwa");
    my $s2pUrvanipAwa = param("s2pUrvanipAwa");
    my $samAsaprakArapUrvanipAwa = param("samAsaprakArapUrvanipAwa");
    my $samAsAnwapUrvanipAwa = param("samAsAnwapUrvanipAwa");
    my $viXayakasUwrapUrvanipAwa =
param("viXayakasUwrapUrvanipAwa");
    my $positionpUrvanipAwa = param("positionpUrvanipAwa");
    my $sUwrapUrvanipAwa = param("sUwrapUrvanipAwa");
    my $dividpUrvanipAwa = param("dividpUrvanipAwa");
    my $anspUrvanipAwa = param("anspUrvanipAwa");

    $p1utf = &cnvrt2utf($p1pUrvanipAwa);
    $p2utf = &cnvrt2utf($p2pUrvanipAwa);
    $avighraha2utf = &cnvrt2utf($avighrahapUrvanipAwa);

    if ($encodingpUrvanipAwa eq "RMN"){
        $p1utfr = &cnvrt2utfr($p1utf);
        $p2utfr = &cnvrt2utfr($p2utf);
        $sUwrautfr = &cnvrt2utfr($sUwrapUrvanipAwa);
    }

    $dividpUrvanipAwa =~ s/^#output//;
```

```

$dividpUrvanipAwa++;

print "<table>";
if ($anspUrvanipAwa eq "No") {$positionpUrvanipAwa = 2;}
if ($anspUrvanipAwa eq "Yes") {$positionpUrvanipAwa = 1;}

if ($positionpUrvanipAwa == 1) {
  if ($encodingpUrvanipAwa eq "RMN"){
    print "<td><font color=\"green\"> pūrvanipātaḥ
$1utf </font></td>";
    print "<td><font color=\"red\"> $sUwrautf </font></td>";
  } else {
    print "<td><font color=\"green\"> पूर्वनिपातः $1utf </font></td>";
    print "<td><font color=\"red\"> $sUwrapUrvanipAwa </font></td>";
  }
}
elseif ($positionpUrvanipAwa == 2) {
  if ($encodingpUrvanipAwa eq "RMN"){
    print "<td><font color=\"green\"> pūrvanipātaḥ
$2utf </font></td>";
    print "<td><font color=\"red\"> $sUwrautf </font></td>";
  } else {
    print "<td><font color=\"green\"> पूर्वनिपातः $2utf </font></td>";
    print "<td><font color=\"red\"> $sUwrapUrvanipAwa </font></td>";
  }
  # Vighraha vAkya with changed order of components
  $avighrahapUrvanipAwa = $p2pUrvanipAwa;
  if($s2pUrvanipAwa ne "") {$avighrahapUrvanipAwa .= "+"
$s2pUrvanipAwa;}
  $avighrahapUrvanipAwa .= " ".$p1pUrvanipAwa;
  if($s1pUrvanipAwa ne "") {$avighrahapUrvanipAwa .= "+"
$s1pUrvanipAwa;}
  $avighraha2utf = &cnvrt2utf($avighrahapUrvanipAwa);

  $tmp = $p1pUrvanipAwa;
  $p1pUrvanipAwa = $p2pUrvanipAwa;
  $p2pUrvanipAwa = $tmp;

  $tmp = $s1pUrvanipAwa;
  $s1pUrvanipAwa = $s2pUrvanipAwa;
  $s2pUrvanipAwa = $tmp;
}
elseif ($positionpUrvanipAwa eq "0") {

```

```

    print "Can not decide the upasarjana saFjFA. <br>";
    print "Compound can not be formed.<br>";
    print "Exiting ...";
    exit;
}
if ($encodingpUrvanipAwa eq "RMN") {
    my $avighraha2utfr = &cnvrt2utfr($avighraha2utf);
    print "<tr><td><font color=\"blue\">$avighraha2utfr</font></td>";
} else {
    print "<tr><td><font color=\"blue\">$avighraha2utf</font></td>";
}

# Note the chaged order of components
my $cmd =
"$myPATH/aluk.out \"$encodingpUrvanipAwa\" \"$avighrapUrvanipAwa\" \"
$plpUrvanipAwa\" \"$s1pUrvanipAwa\" \"$p2pUrvanipAwa\" \"$s2pUrvanipA
wa\" \"$samAsaprakArapUrvanipAwa\" \"$samAsAnwapUrvanipAwa\"
$dividpUrvanipAwa";
system($cmd);
}

```

Aluk.cgi

```
#!/usr/bin/perl

my $myPATH = "/home/pavan/scl/skt_gen/new_compounds";
require "$myPATH/cnvrt2utf.pl";

use warnings;
use CGI ':standard';

my $cgi = new CGI;

print $cgi->header(-type => 'text/html',
                  -charset => 'utf-8');

if (param) {
    my $encodingaluk = param("encodingaluk");
    my $avighrahaaluk = param("vighrahaaluk");
    my $p1aluk = param("p1aluk");
    my $p2aluk = param("p2aluk");
    my $s1aluk = param("s1aluk");
    my $s2aluk = param("s2aluk");
    my $samAsaparakAaaluk = param("samAsaparakAaaluk");
    my $samAsAnwaaluk = param("samAsAnwaaluk");
    my $alukviBARA = param("alukviBARA");
    my $dividaluk = param("dividaluk");
    my $ansaluk = param("ansaluk");

    # aluk = 2: viBARA; aluk + no aluk
    # aluk = 1: aluk
    # aluk = 0: no aluk

    $dividaluk =~ s/#output//;
    $dividaluk++;

    if ($ansaluk ne "Yes") {$alukviBARA = "0";}

    if ($alukviBARA ne "0") { #do aluk kArya
        if ($ansaluk ne "NULL") {
            if ($encodingaluk eq "RMN") {
                my $sUwrautf = &cnvrt2utf($sUwraaluk);
                print "<font color = \"red\"> $sUwrautf </font> <br/>";
            } else {
```

```

        print "<font color = \"red\"> $sUwraaluk </font> <br/>";
    }
}
$cmd =
"$myPATH/alukkArya.out \"$encodingaluk\" \"$p1aluk\" \"$s1aluk\" \"$p2aluk
\" \"$s2aluk\"";
system($cmd);
}
if ($alukviBARA ne "1") { # Do non-aluk kArya
    $cmd =
"$myPATH/nonalukkArya.out \"$encodingaluk\" \"$p1aluk\" \"$s1aluk\" \"$p2
aluk\" \"$s2aluk\" \"$samAsAnwaaluk\" \"$samAsaprakAraaluk\" $dividaluk";
    system($cmd);
}
}

```

Purvapada.cgi
#!/usr/bin/perl

```

my $myPATH = "/home/pavan/scl/skt_gen/new_compounds";
require "$myPATH/cnvrt2utf.pl";
require "$myPATH/cnvrt2utf.pl";

```

```

use warnings;
use CGI ':standard';

```

```

my $cgi = new CGI;

```

```

print $cgi->header(-type => 'text/html',
                  -charset => 'utf-8');

```

```

if (param) {
    my $encodingpUrva = param("encodingpUrva");
    my $p1pUrva = param("p1pUrva");
    my $p2pUrva = param("p2pUrva");
    my $samAsaprakArapUrva = param("samAsaprakArapUrva");
    my $samAsAnwapUrva = param("samAsAnwapUrva");
    my $strmodpUrva = param("strmodpUrva");
    my $strunmodpUrva = param("strunmodpUrva");
    my $sUwrastrpUrva = param("sUwrastrpUrva");
    my $dividpUrva = param("dividpUrva");
    my $anspUrva = param("anspUrva");
}

```

```

#print "Within pUrvapaxa.cgi\n";
#print "samAsAnwa = $samAsAnwapUrva <br />";
$dividpUrva =~ s/#output//;
$dividpUrva++;

if($anspUrva eq "No") {
    $pUrvapaxastr = $strunmodpUrva;
} else {
    $pUrvapaxastr = $strmodpUrva;
}

#print "pUrvapaxa str = $pUrvapaxastr";

my $p1utf = &cnvrt2utf($p1pUrva);
my $pUrvapaxautf = &cnvrt2utf($pUrvapaxastr);

if ($encodingpUrva eq "RMN") {
    my $p1utf = &cnvrt2utf($p1utf);
    my $pUrvapaxautf = &cnvrt2utf($pUrvapaxastrutf);
    my $sUwrastrutf = &cnvrt2utf($sUwrastrpUrva);
}

if ($pUrvapaxastr ne "") {
    if ($encodingpUrva eq "RMN") {
        print "<font color=\"blue\">$p1utf</font> -&gt; <font
color=\"blue\">$pUrvapaxautf</font> <font
color=\"red\">$sUwrastrutf</font>";
    } else {
        print "<font color=\"blue\">$p1utf</font> -&gt; <font
color=\"blue\">$pUrvapaxautf</font> <font
color=\"red\">$sUwrastrpUrva</font>";
    }
    $p1pUrva = $pUrvapaxastr;
}

#printf "p1 = $p1pUrva p2 = $p2pUrva samAsAnwa =
$samAsAnwapUrva<br />";
$cmd =
"$myPATH/uwwarapaxa.out \"$encodingpUrva\" \"$p1pUrva\" \"$p2pUrva\" \"$
samAsAnwapUrva\" \"$samAsaprakArapUrva\" $dividpUrva";
system($cmd);
}

```

Uttarapada.cgi

```
#!/usr/bin/perl

my $myPATH = "/home/pavan/scl/skt_gen/new_compounds";
require "$myPATH/cnvrt2utf.pl";
require "$myPATH/cnvrt2utfr.pl";

use warnings;
use CGI ':standard';

my $cgi = new CGI;

print $cgi->header(-type => 'text/html',
                  -charset => 'utf-8');

if (param) {
    my $encodinguwara = param("encodinguwara");
    my $p1uwara = param("p1uwara");
    my $p2uwara = param("p2uwara");
    my $samAsAnwauwara = param("samAsAnwauwara");
    my $samAsaprakArauwwara = param("samAsaprakArauwwara");
    my $sUwrastruwara = param("sUwrastruwara");
    my $uwwarapaxastr = param("uwwarapaxastr");
    my $dividuwara = param("dividuwara");
    my $ansuwara = param("ansuwara");

    if($ansuwara eq "No") { $uwwarapaxastr = ""; }

    #print "uwwarapaxastr = $uwwarapaxastr\n";
    #print "sUwrastr = $sUwrastruwara\n";
    #print "samAsAnwa = $samAsAnwauwara\n";

    my $p2utf = &cnvrt2utf($p2uwara);
    my $u2utf = &cnvrt2utf($uwwarapaxastr);

    if ($encodinguwara eq "RMN") {
        my $p2utfr = &cnvrt2utfr($p2utf);
        my $u2utfr = &cnvrt2utfr($u2utf);
        my $sUwrastrutfr = &cnvrt2utfr($sUwrastruwara);
    }

    $dividuwara =~ s/#output//;
```

```

$dividuwwara++;

#print "uwwarapaxastr = $uwwarapaxastr\n";
if($uwwarapaxastr ne "") {
    if ($encodinguwwara eq "RMN") {
        print " $p2utfr \&gt; $u2utf <font
color=\"red\">$sUwrastrutf</font><br />";
    } else {
        print " $p2utf \&gt; $u2utf <font
color=\"red\">$sUwrastruwwara</font><br />";
    }
    $p2uwwara = $uwwarapaxastr;
}
#print "Calling paxakArya with samAsAnwa = $samAsAnwauwwara\n";
#print "Calling paxakArya with samAsaprakAra =
$samAsaprakArauwwara\n";
$cmd =
"$myPATH/paxakArya.out \"$encodinguwwara\" \"$p1uwwara\" \"$p2uwwara\"
\" \"$samAsAnwauwwara\" \"$samAsaprakArauwwara\" $dividuwwara";
system($cmd);
}

```

Showindex.cgi

```
#!/usr/bin/perl

my $myPATH="/home/pavan/scl/skt_gen/new_compounds/web_interface/";

use warnings;
use CGI ':standard';

my $cgi = new CGI;

print $cgi->header(-type => 'text/html',
                  -charset => 'utf-8');

if (param) {
    my $encoding = param("encoding");
    my $expert = param("expert");

    if ($expert eq "No") {
        &disply_index($encoding);
    }

    if ($expert eq "Yes") {
        # To add
    }
}

sub display_index {
    my ($encoding) = @_ ;
    if ($encoding eq "RMN") {
        system ("cat $myPATH/index_rmn.html");
    } else {
        system ("cat $myPATH/index.html");
    }
}
1;
```

Samasaparakara.lex

```
%{
#include "struct.h"
extern char sUwrastr[LARGE];
extern char samAsaparakAra[MEDIUM];
}%
%%
^1\4\61    {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"ऊर्यादिच्चिडाचश्च 1.4.61"); }
^1\4\62    {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"अनुकरणं
चानितिपरम् 1.4.62"); }
^1\4\63    {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"आदरानादरयोः सदसती 1.4.63"); }
^1\4\64    {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"भूषणे
अलम् 1.4.64"); }
^1\4\65    {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"अन्तरपरिग्रहे 1.4.65"); }
^1\4\66    {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"कणेमनसी
श्रद्धाप्रतीघाते 1.4.66"); }
^1\4\67    {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"पुरोऽव्ययम् 1.4.67"); }
^1\4\68    {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"अस्तं च
1.4.68"); }
^1\4\69    {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"अच्छगत्यर्थवदेषु 1.4.69"); }
^1\4\70    {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"अदोऽनुपदेशे 1.4.70"); }
^1\4\72    {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"विभाषा
कृञि 1.4.72"); }
^1\4\73    {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"उपाजे
अन्वाजे 1.4.73"); }
^1\4\75    {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"अनत्याधाने उरसिमनसी 1.4.75"); }
^1\4\76    {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"मध्ये पदे
निवचने च 1.4.76"); }
^1\4\77    {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"नित्यं
हस्ते पाणौ उपयमने 1.4.77"); }
^1\4\78    {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"प्राध्वं
बन्धने 1.4.78"); }
^1\4\79    {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"जीविकोपनिषदवौपम्ये 1.4.79"); }
```

^2\1\10 {strcpy(samAsaparakAra,"avyayIBaVaH");
 strcpy(sUwrastr,"अक्षशलाकासख्या परिणा 2.1.10"); }
 ^2\1\12 {strcpy(samAsaparakAra,"avyayIBaVaH");
 strcpy(sUwrastr,"अपपरिबहिरञ्चवः फञ्चया 2.1.12"); }
 ^2\1\13 {strcpy(samAsaparakAra,"avyayIBaVaH"); strcpy(sUwrastr,"आङ्
 मर्यादाभिविध्योः 2.1.13"); }
 ^2\1\14 {strcpy(samAsaparakAra,"avyayIBaVaH");
 strcpy(sUwrastr,"लक्षणेनाभिप्रती आभिमुख्ये 2.1.14"); }
 ^2\1\15 {strcpy(samAsaparakAra,"avyayIBaVaH");
 strcpy(sUwrastr,"अनुर्यत्समया 2.1.15"); }
 ^2\1\16 {strcpy(samAsaparakAra,"avyayIBaVaH"); strcpy(sUwrastr,"यस्य
 चायामः 2.1.16"); }
 ^2\1\18 {strcpy(samAsaparakAra,"avyayIBaVaH"); strcpy(sUwrastr,"पारे
 मध्ये षष्ठ्या वा 2.1.18"); }
 ^2\1\19 {strcpy(samAsaparakAra,"avyayIBaVaH"); strcpy(sUwrastr,"संख्या
 वांश्येन 2.1.19"); }
 ^2\1\20 {strcpy(samAsaparakAra,"avyayIBaVaH");
 strcpy(sUwrastr,"नदीभिश्च 2.1.20"); }
 ^2\1\21 {strcpy(samAsaparakAra,"avyayIBaVaH");
 strcpy(sUwrastr,"अन्यपदार्थे च सञ्ज्ञायाम् 2.1.21"); }
 ^2\1\24 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"द्वितीया
 श्रितातीतपतितगतात्यस्तप्राप्तापन्नैः 2.1.24"); }
 ^2\1\25 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"स्वयम्
 केन 2.1.25"); }
 ^2\1\26 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"खट्वा क्षेपे
 2.1.26"); }
 ^2\1\27 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"सामि
 2.1.27"); }
 ^2\1\28 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"कालाः
 2.1.28"); }
 ^2\1\29 {strcpy(samAsaparakAra,"wawpuruRa");
 strcpy(sUwrastr,"अत्यन्तसंयोगे च 2.1.29"); }
 ^2\1\30 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"तृतीया
 तत्कृतार्थेन गुणवचनेन 2.1.30"); }
 ^2\1\31 {strcpy(samAsaparakAra,"wawpuruRa");
 strcpy(sUwrastr,"पूर्वसदृशसमोनार्थकलहनिपुणमिश्रलक्षणैः 2.1.31"); }
 ^2\1\32 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"कर्तृकरणे
 कृता बहुलम् 2.1.32"); }
 ^2\1\34 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"अन्नेन
 व्यञ्जनम् 2.1.34"); }
 ^2\1\35 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"भक्ष्येण
 मिश्रीकरणम् 2.1.35"); }
 ^2\1\36 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"चतुर्थी

तदर्थार्थबलिहितसुखरक्षितैः 2.1.36"); }

^2\1\37 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"पञ्चमी
भयेन 2.1.37"); }

^2\1\38 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"अपेतापोढमुक्तपतितापत्रस्थैरल्पशः 2.1.38"); }

^2\1\39 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"स्तोकान्तिकदूरर्थकृच्छ्राणि तेन 2.1.39"); }

^2\1\4 {strcpy(samAsaparakAra,""); strcpy(sUwrastr,"सह सुपा 2.1.4"); }

^2\1\40 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"सप्तमी
शौण्डैः 2.1.40"); }

^2\1\41 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"सिद्धशुष्कपक्वबन्धैश्च 2.1.41"); }

^2\1\42 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"ध्वाङ्गेण
क्षेपे 2.1.42"); }

^2\1\43 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"कृत्यैर्ऋणे 2.1.43"); }

^2\1\44 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"सञ्ज्ञायाम्
2.1.44"); }

^2\1\45 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"केनाहोरात्रवयवाः 2.1.45"); }

^2\1\46 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"तत्र
2.1.46"); }

^2\1\47 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"क्षेपे
2.1.47"); }

^2\1\49 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"पूर्वकालैकसर्वजरत्पुराणनवकेवलाः समानाधिकरणेन 2.1.49"); }

^2\1\50 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"दिक्सङ्ख्ये सञ्ज्ञायां 2.1.50"); }

^2\1\52 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"सङ्ख्यापूर्वो द्विगुः 2.1.52"); }

^2\1\53 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"कुत्सितानि कुत्सनैः 2.1.53"); }

^2\1\54 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"पापाणके
कुत्सितैः 2.1.54"); }

^2\1\55 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"उपमाननि
समान्यवचनैः 2.1.55"); }

^2\1\56 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"उपमितं
व्याघ्रादिभिः सामान्याप्रयोगे 2.1.56"); }

^2\1\57 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"विशेषणम्
विशेष्येण बहुलम् 2.1.57"); }

^2\1\58 {strcpy(samAsaparakAra,"wawpuruRa");
strcpy(sUwrastr,"पूर्वापरप्रथमचरमजघन्यसमानमध्यमध्यमवीराश्च 2.1.58"); }

^2\1\6 {strcpy(samAsaparakAra,"avyayIBAvaH"); strcpy(sUwrastr,"अव्ययं

विभक्तिसमीपसमृद्धिव्यूद्ध्यर्थाभावात्ययासम्प्रतिशब्दप्रादुर्भावपश्चाद्यथानुपूर्व्ययौगपद्यसादृश्यसम्पत्तिसाकल्यान्तवचनेषु 2.1.6"); }

^2\1\60 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"केन नञ्विशिष्टेनानञ् 2.1.60"); }

^2\1\61 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"सन्महत्परमोत्तमोत्कृष्टाः पूज्यमानैः 2.1.61"); }

^2\1\62 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"वृन्दारकनागकुञ्जरैः पूज्यमानम् 2.1.62"); }

^2\1\63 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"कतरकतमौ जातिपरिप्रश्ने 2.1.63"); }

^2\1\64 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"किं क्षेपे 2.1.64"); }

^2\1\65 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"पोटायुवतिस्तोककतिपगृष्टिधेनुवशावेहत्बष्कयणीप्रवक्तृश्रोत्रियाध्यापकधूर्तैर् जातिः 2.1.65"); }

^2\1\66 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"प्रशंसावचनैश्च 2.1.66"); }

^2\1\67 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"युवा खलतिपलितवलिनजरतीभिः 2.1.67"); }

^2\1\69 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"वर्णो वर्णेन 2.1.69"); }

^2\1\7 {strcpy(samAsaparakAra,"avyayIBAvaH"); strcpy(sUwrastr,"यथासादृश्ये 2.1.7"); }

^2\1\70 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"कुमारः श्रमणादिभिः 2.1.70"); }

^2\1\71 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"चतुष्पादो गर्भिण्या 2.1.71"); }

^2\1\8 {strcpy(samAsaparakAra,"avyayIBAvaH"); strcpy(sUwrastr,"यावदवधारणे 2.1.8"); }

^2\1\9 {strcpy(samAsaparakAra,"avyayIBAvaH"); strcpy(sUwrastr,"सुप्प्रतिना मात्रार्थे 2.1.9"); }

^2\2\1 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"पूर्वापराधरोत्तरमेकदेशिनैकाधिकरणे 2.2.1"); }

^2\2\10 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"न निर्धारणे 2.2.10"); }

^2\2\17 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"नित्यं क्रीडाजीविकयोः 2.2.17"); }

^2\2\18 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"कुगतिप्रादयः 2.2.18"); }

^2\2\19 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"उपपदमतिङ् 2.2.19"); }

^2\2\2 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"अर्धम्

नपुंसकम् 2.2.2"); }
 ^2\2\20 {strcpy(samAsaparakAra,"wawpuruRa");
 strcpy(sUwrastr,"अमैवाव्ययेन 2.2.20"); }
 ^2\2\21 {strcpy(samAsaparakAra,"wawpuruRa");
 strcpy(sUwrastr,"तृतीयाप्रभृतीन्यन्यतरस्याम् 2.2.21"); }
 ^2\2\22 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"क्त्वा च
 2.2.22"); }
 ^2\2\24 {strcpy(samAsaparakAra,"bahuvrIhi");
 strcpy(sUwrastr,"अनेकमन्यपदार्थे 2.2.24"); }
 ^2\2\27 {strcpy(samAsaparakAra,"bahuvrIhi"); strcpy(sUwrastr,"तत्र
 तेनेदमिति सरूपे 2.2.27"); }
 ^2\2\28 {strcpy(samAsaparakAra,"bahuvrIhi"); strcpy(sUwrastr,"तेन सहेति
 तुल्ययोगे 2.2.28"); }
 ^2\2\29 {strcpy(samAsaparakAra,"xvanxva"); strcpy(sUwrastr,"चार्थे द्वन्द्वः
 2.2.29"); }
 ^2\2\3 {strcpy(samAsaparakAra,"wawpuruRa");
 strcpy(sUwrastr,"द्वितीयतृतीयचतुर्थतुर्याणिअन्यतरस्याम् 2.2.3"); }
 ^2\2\4 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"प्राप्तापन्ने
 च द्वितीयया 2.2.4"); }
 ^2\2\5 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"कालाः
 परिमाणिना 2.2.5"); }
 ^2\2\6 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"नञ्
 2.2.6"); }
 ^2\2\7 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"ईषदकृता
 2.2.7"); }
 ^2\2\8 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"षष्ठी
 2.2.8"); }
 ^2\2\9 {strcpy(samAsaparakAra,"wawpuruRa");
 strcpy(sUwrastr,"याजकादिभिश्च 2.2.9"); }

 ^2\4\5 {strcpy(samAsaparakAra,"xvanxva");
 strcpy(sUwrastr,"अध्ययनतोऽविप्रकृष्टाख्यानाम् 2.4.5"); }
 ^2\4\9 {strcpy(samAsaparakAra,"xvanxva"); strcpy(sUwrastr,"येषां च विरोधः
 शाश्वतिकः 2.4.9"); }

 ^3\1\92 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"तत्रोपपदं
 सप्तमीस्थम् 3.1.92"); }

 ^5\4\69 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"न पूजनात्
 5.4.69"); }
 ^5\4\84 {strcpy(samAsaparakAra,"wawpuruRa"); strcpy(sUwrastr,"द्विस्तावा
 त्रिस्तावा वेदिः 5.4.84"); }

```

^6\3\30    {strcpy(samAsaparakAra,"xvanxva"); strcpy(sUwrastr,"दिवसश्च
पृथिव्याम् 6.3.30"); }
^6\3\32    {strcpy(samAsaparakAra,"xvanxva"); strcpy(sUwrastr,"मातरपितरौ
उदीचाम् 6.3.32"); }
^6\3\25    {strcpy(samAsaparakAra,"xvanxva"); strcpy(sUwrastr,"आनङ्गतो द्वन्द्वे
6.3.25"); }
^6\3\27    {strcpy(samAsaparakAra,"xvanxva"); strcpy(sUwrastr,"ईदग्रेः
सोमवरुणयोः 6.3.27"); }
^6\3\46    {strcpy(samAsaparakAra,"xvanxva"); strcpy(sUwrastr,"आन्महतः
समानाधिकरणजातीययोः 6.3.46"); }
^6\3\48    {strcpy(samAsaparakAra,"xvanxva"); strcpy(sUwrastr,"त्रेस्त्रयः
6.3.48"); }
^6\3\81    {strcpy(samAsaparakAra,"avyayIBAva");
strcpy(sUwrastr,"अव्ययीभावे चाकाले 6.3.81"); }
^.*    { printf("ERROR; %s not covered",yytext);strcpy(samAsaparakAra,"");
strcpy(sUwrastr,""); exit(0);}
%%

```

```

char samAsaparakAra[MEDIUM];
char sUwrastr[LARGE];
void print_samAsaparakAra();

```

```

#define myPATH "/home/pavan/scl/skt_gen/new_compounds/"

```

```

main (int argc, char *argv[]) {
char cmd[MEDIUM];
char avigraha[MEDIUM];
char samAsaparakAra_utf[MEDIUMUTF];
char samAsaparakAra_utfr[MEDIUMUTF];
char avigraha_utf[MEDIUMUTF];
char avigraha_utfr[MEDIUMUTF];
char sUwrastr_utfr[MEDIUMUTF];
char sUwrasaMKyA[SMALL];
char encoding[MEDIUM];

```

```

/* argv[1] : encoding
   argv[2] : avigraha
   argv[3] : prAwipaxikam1
   argv[4] : sup1
   argv[5] : prAwipaxikam2
   argv[6] : sup2
   argv[7] : sUwrasaMKyA
*/

```

```
strcpy(encoding,argv[1]);
strcpy(avigraha,argv[2]);
strcpy(sUwrasaMKyA,argv[7]);

//cnvrtwx2utf(argv[2],in);

//printf("<center><h4><font color=\"blue\">");
//printf("%s",in);
//printf(" (%s) </font></h4></center>\n",instring);

call_samAsaprakAralex(sUwrasaMKyA);

cnvrtwx2utf(samAsaprakAra,samAsaprakAra_utf);
cnvrtwx2utf(avigraha,avigraha_utf);

if (!strcmp(encoding,"RMN")){

    cnvrtutfd2r(avigraha_utf,avigraha_utfr);
    cnvrtutfd2r(samAsaprakAra_utf,samAsaprakAra_utfr);
    cnvrtutfd2r(sUwrastr,sUwrastr_utfr);

    print_samAsaprakAra(avigraha_utfr, "samāsaprakārah",
samAsaprakAra_utfr,sUwrastr_utfr);
} else {

print_samAsaprakAra(avigraha_utf,"समासप्रकार:",samAsaprakAra_utf,sUwrastr
);
}

sprintf(cmd,"%s/samAsAnwa.out \"%s\" \"%s\" \"%s\" \"%s\" \"%s\" \"%s\" \"%s\" \"%s\" \"%s\"",myPATH,
encoding,avigraha,argv[3],argv[4],argv[5],argv[6],sUwrasaMKyA,samAsaprak
Ara);
system(cmd);
}
call_samAsaprakAralex(char *in){
    yy_scan_string(in);
    yylex();
}
yywrap()
{
}
```

```

void print_samAsaparakAra (char *avigraha, char *title, char *samAsaparakAra,
char *sUwra)
{
    printf("<table><tr>");
    printf("<td><font color=\"blue\"> %s", avigraha);
    printf("</font></td>");
    printf("<td><font color=\"green\">%s %s", title, samAsaparakAra);
    printf("</font></td>");
    printf("<td><font color=\"red\"> %s </font></td>", sUwra);
    fflush(stdout);
}

```

Samaasaanta.lex

```
%{
#include "struct.h"
extern char ques[LARGE];
extern char samAsaprakAra[MEDIUM];
extern char p1_utf[MEDIUMUTF];
extern char p2_utf[MEDIUMUTF];
}%

prA [a-zA-Z]+
sup \(su|O|jas|am|Ot|Sas|tA|Bis|fe|fasi|ByAm|Byas|fas|os6|Am|fi|os7|sup)

prAWamA (su|O|jas)
word {prA} {sup}

avy216      (upa|aXi|anu|nir|apa|xur|awi|praw|pari|yAvax|A)
pUraNI (prAWamA|xviwIyA|wqwIyA|cawurWI|paFcami|RaRTI|sapwami|
aRTami|navami|xaSamI)
safKyA      (ekam|xve|wrINi|cawvAri|paFca|RaRTa|sapwa|aRTa|nava|xaSa|
ekAxaSa|xvAxaSa|wrayoxaSa|cawurxaSa|paFcaxaSa|RodaSa|sapwaxaSa|
aRTAxaSa|navaxaSa|viMSawi|ekaviMSawi|xvAviMSawi|wrayoviMSawi|
cawurviMSaWi|paFcaviMSawi|RadviMSaWi|sapwaviMSawi|aRtAviMSawi|
navaviMSawi|wriMSaw|ekawriMSaw|xvAwriMSaw|wrayaswriMSaw|
cawuswriMSaw|paFcawriMSaw|RatwriMSaw|sapwawriMSaw|aRTAwriMSaw|
navawriMSaw|cawvAriMSaw|ekacawvAriMSaw|xvicawvAriMSaw|
wricawvAriMSaw|cawuscawvAriMSaw|paFcacawvAriMSaw|
RatcawvAriMSaw|sapwacawvAriMSaw|aRtAcawvAriMSaw|
navacawvAriMSaw|paFcASaW)
saMKyeyAbahugaNa (ekam|xve|wrINi|cawvAri|paFca|RaRTa|sapwa|aRTa|
nava|xaSa|ekAxaSa|xvAxaSa|wrayoxaSa|cawurxaSa|paFcaxaSa|RodaSa|
sapwaxaSa|aRTAxaSa|navaxaSa|viMSawi|ekaviMSawi|xvAviMSawi|
wrayoviMSawi|cawurviMSaWi|paFcaviMSawi|RadviMSaWi|sapwaviMSawi|
aRtAviMSawi|navaviMSawi|wriMSaw|ekawriMSaw|xvAwriMSaw|
wrayaswriMSaw|cawuswriMSaw|paFcawriMSaw|RatwriMSaw|
sapwawriMSaw|aRTAwriMSaw|navawriMSaw|cawvAriMSaw|
ekacawvAriMSaw|xvicawvAriMSaw|wricawvAriMSaw|cawuscawvAriMSaw|
paFcacawvAriMSaw|RatcawvAriMSaw|sapwacawvAriMSaw|
aRtAcawvAriMSaw|navacawvAriMSaw|paFcASaW)
SarawpraBqwi (Sarax|vipAS|anas|manas|upAnah|xiv|himavaw|anaduh|xiS|
xqS|viS|cewas|cawur|wyax|wax|yax|kiyaw)
uraHpraBqwi (uras|sarpis|upAnah|pumAn|anadvAn|payaH|nOH|lakRmIH|
xaXimaXu|SAlI|SAlIH)
```

%%

```
^kim[ ]{word}[ ].*      {
    sprintf(ques,"किम् इति निन्दार्थे वर्तते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"","किमः क्षेपे 5.4.70");
} /* kim rAjan+su */

^{word}[ ]kim[ ].*      {
    sprintf(ques,"किम् इति निन्दार्थे वर्तते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"","किमः क्षेपे 5.4.70");
} /* kim rAjan+su */

^na[ ]paWin{sup}[ ].*    { samAsAnwa_cgi_interface("NULL","ac","पथो
विभाषा 5.4.72");}
^paWin{sup}[ ]na[ ].*    { samAsAnwa_cgi_interface("NULL","ac","पथो
विभाषा 5.4.72");}
^na[ ]{word}[ ]wawpuruRa    {

samAsAnwa_cgi_interface("NULL","", "नञस्तत्पुरुषात् 5.4.71");
} /* na rAjan+su */
^{word}[ ]na[ ]wawpuruRa    {

samAsAnwa_cgi_interface("NULL","", "नञस्तत्पुरुषात् 5.4.71");
} /* na rAjan+su */
^{saMKyeyAbahugana}{sup}[ ]{word}[ ]bahuvrIhi    {
    sprintf(ques," %s इति पदम् संख्येयार्थे अस्ति
(Y/N)?", p1_utf);

samAsAnwa_cgi_interface(ques,"dac","बहुव्रीहो संख्येये डजबहुगणात् 5.4.73");
}
^{word}[ ]{saMKyeyAbahugana}{sup}[ ]bahuvrIhi    {
    sprintf(ques," %s इति पदम् संख्येयार्थे अस्ति
(Y/N)?", p1_utf);

samAsAnwa_cgi_interface(ques,"dac","बहुव्रीहो संख्येये डजबहुगणात् 5.4.73");
}
^akRa{sup}[ ]Xur{sup}[ ].*
    { samAsAnwa_cgi_interface("NULL","", "ऋक्पूरब्धूः पथामानक्षे 5.4.74"); }
^Xur{sup}[ ]akRa{sup}[ ].*
    { samAsAnwa_cgi_interface("NULL","", "ऋक्पूरब्धूः पथामानक्षे 5.4.74"); }
/* akRa+fas Xur+su (2.2.8) +su */
^{( {word}|pra|para|apa|sam|anu|ava|nis|nir|xus|xur|vi|A|ni|aXi|api|awi|su|uw|aBi|
prawi|pari|upa)[ ](qk|ap|pur|Xur|paWin){sup}[ ].*
    {samAsAnwa_cgi_interface("", "ap", "ऋक्पूरब्धूः पथामानक्षे 5.4.74"); }
^(qk|ap|pur|Xur|paWin){sup}[ ]( {word}|pra|para|apa|sam|anu|ava|nis|nir|xus|xur|
```

```

vi|A|ni|aXi|api|awi|su|uw|aBi|prawi|pari|upa)[ ].*
{samAsAnwa_cgi_interface("", "ap", "ऋक्पूरब्धूः पथामानक्षे 5.4.74"); }
^(prawi|anu|ava)[ ](sAman|loman){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL", "ac", "अच्छप्रत्यन्ववपूर्वत्सामलोम्नः
5.4.75"); }
^(sAman|loman)[ ](prawi|anu|ava){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL", "ac", "अच्छप्रत्यन्ववपूर्वत्सामलोम्नः
5.4.75"); }
^(na|vi|su)[ ]cawur{sup}[ ].*
^cawur{sup}[ ](na|vi|su)[ ].*
^swrI{sup}[ ]puMs{sup}[ ].*
^puMs{sup}[ ]swrI{sup}[ ].*
^Xenu{sup}[ ]anaduh{sup}[ ].*
^anaduh{sup}[ ]Xenu{sup}[ ].*
^qc{sup}[ ]sAman{sup}[ ].*
^sAman{sup}[ ]qc{sup}[ ].*
^vAc{sup}[ ]manas{sup}[ ].*
^manas{sup}[ ]vAc{sup}[ ].*
^akRi{sup}[ ]BrU{sup}[ ].*
^BrU{sup}[ ]akRi{sup}[ ].*
^xAra{sup}[ ]go{sup}[ ].*
^go{sup}[ ]xAra{sup}[ ].*
^UrU{sup}[ ]aRTIvaw{sup}[ ].*
^aRTIvaw{sup}[ ]UrU{sup}[ ].*
^pAxa{sup}[ ]aRTIvaw{sup}[ ].*
^aRTIvaw{sup}[ ]pAxa{sup}[ ].*
^xivA[ ]nakwa[ ].*
^nakwa[ ]xivA[ ].*
^rAwri{sup}[ ]xivA{sup}[ ].*
^xivA{sup}[ ]rAwri{sup}[ ].*
^ahan{sup}[ ]xivA{sup}[ ].*
^xivA{sup}[ ]ahan{sup}[ ].*
^rajas{sup}[ ]saha[ ].*
^saha[ ]rajas{sup}[ ].*
^nis[ ]Sreyas{sup}[ ].*
^Sreyas{sup}[ ]nis[ ].*
^puruRa{sup}[ ]Ayus{sup}[ ].*
^Ayus{sup}[ ]puruRa{sup}[ ].*
^(xvi|wri){sup}[ ]Ayus{sup}[ ].*
^Ayus{sup}[ ](xvi|wri){sup}[ ].*
^qc{sup}[ ]yajus{sup}[ ].*
^yajus{sup}[ ]qc{sup}[ ].*
^(jAwa|mahAn|vqxXa){sup}[ ]ukRan{sup}[ ].*

```

```

^ukRan{sup}[ ](jAwa|mahAn|vqxXa){sup}[ ].* |
^go{sup}[ ]Svan{sup}[ ].* |
^Svan{sup}[ ]go{sup}[ ].* |
^Svan{sup}[ ]upa[ ].* |
^upa[ ]Svan{sup}[ ].*

{samAsAnwa_cgi_interface("NULL","ac","अचतुरविचतुरसुचतुरस्त्रीपुंसधेन्वनडुहर्क्स
ामवाङ्मनसाक्षिभ्रुवदारगवोर्वष्ठीवपदष्ठीवनक्तन्दिवरात्रिन्दिवाहर्दिवसरजसनिःश्रेयसपुरुषायुषद्
यायुषत्र्यायुषर्ग्यजुषजातोक्षमहोक्षवृद्धोक्षोपशुनगोष्ठष्वाः 5.4.77"); }
^{word}[ ]akRi{sup}[ ].* {
    sprintf(ques,"किम् अक्षिशब्दः अदर्शनार्थे वर्तते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"ac","अक्षणोऽदर्शनात्
5.4.76");
}
^akRi{sup}[ ]{word}[ ].* {
    sprintf(ques,"किम् अक्षिशब्दः अदर्शनार्थे वर्तते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"ac","अक्षणोऽदर्शनात्
5.4.76");
}
^varcas{sup}[ ](brahman|haswin){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","ब्रह्महस्तिभ्यां वर्चसः 5.4.78"); }
^(brahman|haswin){sup}[ ]varcas{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","ब्रह्महस्तिभ्यां वर्चसः 5.4.78"); }
^(ava|sam|anXa){sup}[ ]wamas{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","अवसमन्धेभ्यस्तमसः
5.4.79"); }
^wamas{sup}[ ](ava|sam|anXa){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","अवसमन्धेभ्यस्तमसः
5.4.79"); }
^Svas{sup}[ ](vasIyas|Sreyas){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","श्वसो वसीयःश्रेयसः 5.4.80"); }
^(vasIyas|Sreyas){sup}[ ]Svas{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","श्वसो वसीयःश्रेयसः 5.4.80"); }
^wapwa{sup}[ ]rahas{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","अन्ववतप्ताद्रहसः 5.4.81"); }
^rahas{sup}[ ]wapwa{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","अन्ववतप्ताद्रहसः 5.4.81"); }
^(anu|ava)[ ]rahas{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","अन्ववतप्ताद्रहसः 5.4.81"); }
^rahas{sup}[ ](anu|ava)[ ][ ].*
    {samAsAnwa_cgi_interface("NULL","ac","अन्ववतप्ताद्रहसः 5.4.81"); }
^prawi[ ]uras{sup}[ ].* {samAsAnwa_cgi_interface("NULL","ac","प्रतेरुरसः
सप्तमीस्थात् 5.4.82"); }

```

```

^uras[ ]prawi{sup}[ ].* {samAsAnwa_cgi_interface("NULL","ac","प्रतेरुसः
सप्तमीस्थात् 5.4.82"); }
^anu[ ]go{sup}[ ]avyayIBAvaH[ ].* {
    sprintf(ques,"अनु इति अव्ययम् आयामर्थे वर्तते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"ac","अनुगवमायामे
5.4.83");
}
^go{sup}[ ]anu[ ]avyayIBAvaH[ ].* {
    sprintf(ques,"अनु इति अव्ययम् आयामर्थे वर्तते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"ac","अनुगवमायामे
5.4.83");
}
^(xvis|wris)[ ]wAvawI{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","द्विस्तावात्रिस्तावा वेदिः 5.4.84");
}
^wAvawI{sup}[ ](xvis|wris)[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","द्विस्तावात्रिस्तावा वेदिः 5.4.84");
}
^(pra|parA|apa|sam|anu|ava|nis|nir|xus|xur|vi|Af|ni|aXi|awi|api|su|ux|aBi|prawi|
para|upa)[ ]aXvan{sup}[ ].*
    {samAsAnwa_cgi_interface("", "ac", "उपसर्गादिध्वनः 5.4.85"); }
^aXvan{sup}[ ](pra|parA|apa|sam|anu|ava|nis|nir|xus|xur|vi|Af|ni|aXi|awi|api|su|
ux|aBi|prawi|para|upa)[ ].*
    {samAsAnwa_cgi_interface("", "ac", "उपसर्गादिध्वनः 5.4.85"); }
^{safKyA}{sup}[ ]afguli{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","तत्पुरुषस्याङ्गुलेः सङ्ख्याव्ययादेः
5.4.86"); }
^afguli{sup}[ ]{safKyA}{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","तत्पुरुषस्याङ्गुलेः सङ्ख्याव्ययादेः
5.4.86"); }
^(ahan|sarva|saMKyAwa|puNya|{safKyA}){sup}[ ]rAwri{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","अहस्सर्वैकदेशसख्यातपुण्याच्च
रात्रेः 5.4.87"); }
^rAwri{sup}[ ](ahan|sarva|saMKyAwa|puNya|{safKyA}){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","अहस्सर्वैकदेशसख्यातपुण्याच्च
रात्रेः 5.4.87"); }
^(rAjan|ahan|saKi)\+{praWamA}[ ]{word}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","राजाहस्सखिभ्यष्टच् 5.4.91");}
^{word}[ ](rAjan|ahan|saKi)\+{praWamA}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","राजाहस्सखिभ्यष्टच् 5.4.91");}
/* 5.4.92 -- taddhita pratyaya */
^uras{sup}[ ]{word}[ ]wawpuruRa {
    sprintf(ques,"अनेन समासेन मुख्यार्थः उच्यते(Y/N)?");

```

```

        samAsAnwa_cgi_interface(ques,"tac","अग्राख्यायामुरसः
5.4.93");
    }
    ^{word}[ ]uras{sup}[ ]wawpuruRa {
        sprintf(ques,"अनेन समासेन मुख्यार्थः उच्यते(Y/N)?");
        samAsAnwa_cgi_interface(ques,"tac","अग्राख्यायामुरसः
5.4.93");
    }
    ^{word}[ ](anas|aSman|aingyas|saras){sup}[ ]wawpuruRa {
        sprintf(ques,"अनेन समासेन जातिः
अथवा संज्ञा गम्यते(Y/N)?");
        samAsAnwa_cgi_interface(ques,"tac","अनोश्मायस्सरसां जातिसंज्ञयोः 5.4.94");
    }
    ^((anas|aSman|aingyas|saras){sup}[ ]){word}[ ]wawpuruRa {
        sprintf(ques,"अनेन समासेन जातिः
अथवा संज्ञा गम्यते(Y/N)?");
        samAsAnwa_cgi_interface(ques,"tac","अनोश्मायस्सरसां जातिसंज्ञयोः 5.4.94");
    }
    ^((grAma|kOta){sup}[ ]wakRan{sup}[ ])*
        {samAsAnwa_cgi_interface("NULL","tac","ग्रामकौटाभ्यां च तक्षणः 5.4.95");
    }
    ^wakRan{sup}[ ]((grAma|kOta){sup}[ ])*
        {samAsAnwa_cgi_interface("NULL","tac","ग्रामकौटाभ्यां च तक्षणः 5.4.95");
    }
    ^awi[ ]Svan{sup}[ ]*
        {samAsAnwa_cgi_interface("NULL","tac","अतेः शुनः 5.4.96"); }
    ^Svan{sup}[ ]awi[ ]*
        {samAsAnwa_cgi_interface("NULL","tac","अतेः शुनः 5.4.96"); }
    ^((su|awi)[ ]){word}[ ]wawpuruRaH {
        sprintf(ques,"%s इति पदं पूजार्थे वर्तते(Y/N)?",
p1_utf);
        samAsAnwa_cgi_interface(ques," ","न पूजनात्
5.4.69");
    }
    ^{word}[ ]((su|awi)[ ]wawpuruRaH {
        sprintf(ques,"%s इति पदं पूजार्थे वर्तते(Y/N)?",
p1_utf);
        samAsAnwa_cgi_interface(ques," ","न पूजनात्
5.4.69");
    }
    /* su+su rAjan+su */

```

```

^sakWi{sup}[ ](uwwara|mqga|pUrva|Palaka){sup}[ ].*
    { samAsAnwa_cgi_interface("NULL","tac","उत्तरमृगपूर्वाच्च सक्थनः
5.4.98");}
^(uwwara|mqga|pUrva|Palaka){sup}[ ]sakWi{sup}[ ].*
    { samAsAnwa_cgi_interface("NULL","tac","उत्तरमृगपूर्वाच्च सक्थनः
5.4.98");}
^(eka|xvi|wri){sup}[ ]nO{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","नावो द्विगोः 5.4.99"); }
^nO{sup}[ ](eka|xvi|wri){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","नावो द्विगोः 5.4.99"); }
^nO{sup}[ ]arXa{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","अर्धाच्च 5.4.100");}
^arXa{sup}[ ]nO{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","अर्धाच्च 5.4.100");}
^(eka|xvi|wri|arXa){sup}[ ]KArI{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","खार्याः प्राचाम् 5.4.101"); }
^KArI{sup}[ ](eka|xvi|wri|arXa){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","खार्याः प्राचाम् 5.4.101"); }
^(xvi|wri){sup}[ ]aFjali{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","द्वित्रिभ्यामञ्जलेः 5.4.102"); }
^aFjali{sup}[ ](xvi|wri){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","द्वित्रिभ्यामञ्जलेः 5.4.102"); }
/* 5.4.103 -- Canxasi */
^ku[ ]brahman{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","कुमहद्भ्यामन्यतरस्याम्
5.4.105"); }
^brahman{sup}[ ]ku[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","कुमहद्भ्यामन्यतरस्याम्
5.4.105"); }
^{word}[ ]brahman{sup}[ ]wawpuruRa    {
    sprintf(ques,"अनेन समासेन जानपदाख्यार्थः
उच्यते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"tac","ब्रह्मणो
जानपदाख्यायाम् 5.4.104");
    }
^brahman{sup}[ ]{word}[ ]wawpuruRa    {
    sprintf(ques,"अनेन समासेन जानपदाख्यार्थः
उच्यते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"tac","ब्रह्मणो
जानपदाख्यायाम् 5.4.104");
    }
^(ku|mahaw){sup}[ ]brahman{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","कुमहद्भ्यामन्यतरस्याम्

```

```

5.4.105"); }
^brahman{sup}[ ](ku|mahaw){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","कुमहद्भ्यामन्यतरस्याम्
5.4.105"); }
^{\word}[ ]{\prA}[cCjJFxRh]\+{\praWamA}[ ]xvanxva {

samAsAnwa_cgi_interface("NULL","tac","द्वन्द्वाच्चुदषहान्तात्समाहारे 5.4.106");
    }
^{\prA}[cCjJFxRh]\+{\praWamA}[ ]{\word}[ ]xvanxva {

samAsAnwa_cgi_interface("NULL","tac","द्वन्द्वाच्चुदषहान्तात्समाहारे 5.4.106");
    }
^(aXi|upa)[ ]{\prA}an{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","अनश्च 5.4.108"); }
^{\prA}an{sup}[ ](aXi|upa)[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","अनश्च 5.4.108"); }
^upa[ ](naxI|pOrNamAsI|AgrahAyaNI){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","नदीपौउर्णमाश्याग्रहायणीभ्यः
5.4.110"); }
^(naxI|pOrNamAsI|AgrahAyaNI){sup}[ ]upa[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","नदीपौउर्णमाश्याग्रहायणीभ्यः
5.4.110"); }
^{\SarawpraBqwi}{sup}[ ]{\avy216}[ ]avyayIBAvaH
    {samAsAnwa_cgi_interface("NULL","tac","अव्ययीभावे शस्त्रभृतिभ्यः
5.4.107"); }
^{\avy216}[ ]{\SarawpraBqwi}{sup}[ ]avyayIBAvaH
    {samAsAnwa_cgi_interface("NULL","tac","अव्ययीभावे शस्त्रभृतिभ्यः
5.4.107"); }
^(upa|pra|para|ava|sam)[ ]{\prA}[JBGDxjbgdxKPCTWcwkp]{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","झयः 5.4.111"); }
^{\prA}[JBGDxjbgdxKPCTWcwkp]{sup}[ ](upa|pra|para|ava|sam)[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","झयः 5.4.111"); }
^upa[ ]giri{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","गिरेश्च सेनकस्य 5.4.112"); }
^giri{sup}[ ]upa[ ].*
    {samAsAnwa_cgi_interface("NULL","tac","गिरेश्च सेनकस्य 5.4.112"); }
^{\word}[ ](sakWi|akRi){sup}[ ]bahuvrIhi {
    sprintf(ques,"अनेन समासेन स्वाङ्गार्थः
उच्यते(Y/N)?");

samAsAnwa_cgi_interface(ques,"Rac","बहुव्रीहौ सक्थ्यक्ष्णौः स्वाङ्गात् षच् 5.4.113");
    }
^(sakWi|akRi){sup}[ ]{\word}[ ]bahuvrIhi {

```

sprintf(ques,"अनेन समासेन स्वाङ्गार्थः

उच्यते(Y/N)?");

samAsAnwa_cgi_interface(ques,"Rac","बहुव्रीहौ सक्थ्यक्षणौः स्वाङ्गात् षच् 5.4.113");
}

^{word}[]afguli{sup}[].*
{samAsAnwa_cgi_interface("NULL","Rac","अङ्गुलेर्दारुणि 5.4.114"); }

^afguli{sup}[]{word}[].*
{samAsAnwa_cgi_interface("NULL","Rac","अङ्गुलेर्दारुणि 5.4.114"); }

^(xvi|wri){sup}[]mUrXan{sup}[].*
{samAsAnwa_cgi_interface("NULL","Ra","द्वित्रिभ्यां ष मूर्ध्नः 5.4.115"); }

^mUrXan{sup}[](xvi|wri){sup}[].*
{samAsAnwa_cgi_interface("NULL","Ra","द्वित्रिभ्यां ष मूर्ध्नः 5.4.115"); }

^{word}[](pramANI|pUraNI){sup}[]bahuvrIhi
{samAsAnwa_cgi_interface("NULL","ap","अप् पूरणीप्रमाण्योः 5.4.116"); }

^(pramANI|pUraNI){sup}[]{word}[]bahuvrIhi
{samAsAnwa_cgi_interface("NULL","ap","अप् पूरणीप्रमाण्योः 5.4.116"); }

^(anwar|bahis)[]loman{sup}[].*
{samAsAnwa_cgi_interface("NULL","ap","अन्तर्बहिर्भ्याम् च लोम्नः
5.4.117"); }

^loman{sup}[](anwar|bahis)[].*
{samAsAnwa_cgi_interface("NULL","ap","अन्तर्बहिर्भ्याम् च लोम्नः
5.4.117"); }

^sWulA{sup}[]nAsikA{sup}[]bahuvrIhi
{samAsAnwa_cgi_interface("NULL","", "अञ्जासिकायाः संज्ञायाम् नसं
चास्थुलात् 5.4.118"); }

^nAsikA{sup}[]sWulA{sup}[]bahuvrIhi
{samAsAnwa_cgi_interface("NULL","", "अञ्जासिकायाः संज्ञायाम् नसं
चास्थुलात् 5.4.118"); }

^{word}[]nAsikA{sup}[]bahuvrIhi {

samAsAnwa_cgi_interface("NULL","ac","अञ्जासिकायाः संज्ञायाम् नसं चास्थुलात्
5.4.118");

}

^nAsikA{sup}[]{word}[]bahuvrIhi {

samAsAnwa_cgi_interface("NULL","ac","अञ्जासिकायाः संज्ञायाम् नसं चास्थुलात्
5.4.118");

}

^(pra|para|apa|sam|anu|ava|nis|nir|xus|xur|vi|Af|ni|aXi|awi|api|su|ux|aBi|prawi|
pari|upa)[]nAsikA\+{praWamA}[].*

{samAsAnwa_cgi_interface("NULL","ac","उपसर्गाच्च 5.4.119"); }

^nAsikA\+{praWamA}[](pra|para|apa|sam|anu|ava|nis|nir|xus|xur|vi|Af|ni|aXi|

```

awi|api|su|ux|aBi|prawi|pari|upa)[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","उपसर्गाच्च 5.4.119"); }
^(prAwar|xivA|Svas|kukRi|aSri|paxa){sup}[ ](su|SAri|cawur|eNI|aja|proRTa)
{sup}[ ].*

{samAsAnwa_cgi_interface("NULL","ac","सुप्रातसुश्वसुदिवशारिकुक्षचतुरश्रैणिपदाज
पदप्रोष्ठपदा: 5.4.120"); }
^(su|SAri|cawur|eNI|aja|proRTa){sup}[ ](prAwar|xivA|Svas|kukRi|aSri|paxa)
{sup}[ ].*

{samAsAnwa_cgi_interface("NULL","ac","सुप्रातसुश्वसुदिवशारिकुक्षचतुरश्रैणिपदाज
पदप्रोष्ठपदा: 5.4.120"); }
^(a|xus|su)[ ](hali|sakWi){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","नञ्दुःसुभ्यो
हलिसक्थ्योरन्यतस्याम् 5.4.121"); }
^(hali|sakWi){sup}[ ](a|xus|su)[ ].*
    {samAsAnwa_cgi_interface("NULL","ac","नञ्दुःसुभ्यो
हलिसक्थ्योरन्यतस्याम् 5.4.121"); }
^(a|xus|su)[ ](prajA|meXA){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","asic","नित्यमसिचप्रजामेधयोः
5.4.122"); }
^(prajA|meXA){sup}[ ](a|xus|su)[ ].*
    {samAsAnwa_cgi_interface("NULL","asic","नित्यमसिचप्रजामेधयोः
5.4.122"); }
/* 5.4.123 -- Canxasi */
^{word}[ ]Xarma{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","anic","धर्मादिनिच्केवलात्
5.4.124"); }
^Xarma{sup}[ ]{word}[ ].*
    {samAsAnwa_cgi_interface("NULL","anic","धर्मादिनिच्केवलात्
5.4.124"); }
^(su|hariwa|wqNa|soma){sup}[ ]jamBa{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","anic","जम्भासुहरिततृणसोमेभ्यः
5.4.125"); }
^jamBa{sup}[ ](su|hariwa|wqNa|soma){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","anic","जम्भासुहरिततृणसोमेभ्यः
5.4.125"); }
^xakRiNa{sup}[ ]Irma{sup}[ ]bahuvrIhi
                                                                    {
                                                                    sprintf(ques,"अनेन समासेन लुब्धयोगः
उच्यते(Y/N)?");
                                                                    }

samAsAnwa_cgi_interface(ques,"anic","दक्षिणेर्मा लुब्धयोगे 5.4.126");
                                                                    }

```

```

^Irma{sup}[ ]xakRiNa{sup}[ ]bahuvrIhi
                                {
                                sprintf(ques,"अनेन समासेन लुब्धयोगः
उच्यते(Y/N)?");

samAsAnwa_cgi_interface(ques,"anic","दक्षिणेर्मा लुब्धयोगे 5.4.126");
                                }
^XanuR{sup}[ ]{word}[ ]bahuvrIhi
                                {
                                sprintf(ques,"अनेन समासेन संज्ञा गम्यते(Y/N)?");
                                samAsAnwa_cgi_interface(ques,"anaf","वा
संज्ञायां 5.4.133");
                                }
^ {word}[ ]XanuR{sup}[ ]bahuvrIhi
                                {
                                sprintf(ques,"अनेन समासेन संज्ञा गम्यते(Y/N)?");
                                samAsAnwa_cgi_interface(ques,"anaf","वा
संज्ञायां 5.4.133");
                                }
^xaNda{sup}[ ]xvi{sup}[ ]bahuvrIhi
    {samAsAnwa_cgi_interface("NULL","ic","द्विदण्ड्याडिभ्यश्च 5.4.128");}
^xvi{sup}[ ]xaNda{sup}[ ]bahuvrIhi
    {samAsAnwa_cgi_interface("NULL","ic","द्विदण्ड्याडिभ्यश्च 5.4.128");}
^UXas{sup}[ ]{word}[ ]bahuvrIhi
    {samAsAnwa_cgi_interface("NULL","anaf","ऊधसोऽनङ् 5.4.131");}
^ {word}[ ]UXas{sup}[ ]bahuvrIhi
    {samAsAnwa_cgi_interface("NULL","anaf","ऊधसोऽनङ् 5.4.131");}
^XanuR{sup}[ ]{word}[ ].*
    {samAsAnwa_cgi_interface("NULL","anaf","धनुषश्च 5.4.132"); }
^ {word}[ ]XanuR{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","anaf","धनुषश्च 5.4.132"); }
^ {word}[ ]jAyA{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","nif","जायाय निङ् 5.4.134"); }
^jAyA{sup}[ ]{word}[ ].*
    {samAsAnwa_cgi_interface("NULL","nif","जायाय निङ् 5.4.134"); }
^ganXa{sup}[ ](pUwi|suraBi){sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","i","गन्धस्येदुत्पूतिसुसुरभिभ्यः
5.4.135"); }
^(pUwi|suraBi){sup}[ ]ganXa{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","i","गन्धस्येदुत्पूतिसुसुरभिभ्यः
5.4.135"); }
^ganXa{sup}[ ](uw|su)[ ].*
    {samAsAnwa_cgi_interface("NULL","i","गन्धस्येदुत्पूतिसुसुरभिभ्यः
5.4.135"); }
^(uw|su)[ ]ganXa{sup}[ ].*
    {samAsAnwa_cgi_interface("NULL","i","गन्धस्येदुत्पूतिसुसुरभिभ्यः

```

```

5.4.135"); }
^ganXa{sup}[ ]sUpa{sup}[ ]bahuvrIhi {
    sprintf(ques,"अनेन समासेन अल्पार्थः
गम्यते(Y/N)?");

samAsAnwa_cgi_interface(ques,"i","अल्पाख्यायाम् 5.4.136");
}
^sUpa{sup}[ ]ganXa{sup}[ ]bahuvrIhi {
    sprintf(ques,"अनेन समासेन अल्पार्थः
गम्यते(Y/N)?");

samAsAnwa_cgi_interface(ques,"i","अल्पाख्यायाम् 5.4.136");
}
^ganXa{sup}[ ]{word}[ ]bahuvrIhi {
    sprintf(ques,"किम् %s पदम् %s पदस्य उपमानं
वा(Y/N)?", p1_utf,p2_utf);
    samAsAnwa_cgi_interface(ques,"i","उपमानाच्च
5.4.137");
}
^{word}[ ]ganXa{sup}[ ]bahuvrIhi {
    sprintf(ques,"किम् %s पदम् %s पदस्य उपमानं
वा(Y/N)?", p1_utf,p2_utf);
    samAsAnwa_cgi_interface(ques,"i","उपमानाच्च
5.4.137");
}
/* 5.4.142 -- Canxasi */
^{uraHpraBqwi}{sup}[ ]{word}[ ]bahuvrIhi
    {samAsAnwa_cgi_interface("NULL","kap","उरःप्रभृतिभ्यः कप् 5.4.151");}
^{word}[ ]{uraHpraBqwi}{sup}[ ]bahuvrIhi
    {samAsAnwa_cgi_interface("NULL","kap","उरःप्रभृतिभ्यः कप् 5.4.151");}
^{prA}in{sup}[ ]{word}[ ]bahuvrIhi {
    sprintf(ques,"किम् अनेन समासेन स्त्रीत्वं
द्योत्यते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"kap","इनः
स्त्रियाम् 5.4.152");
}
^{word}[ ]{prA}in{sup}[ ]bahuvrIhi {
    sprintf(ques,"किम् अनेन समासेन स्त्रीत्वं
द्योत्यते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"kap","इनः
स्त्रियाम् 5.4.152");
}
^pravANI{sup}[ ]nir[ ]bahuvrIhi {

```

```

        samAsAnwa_cgi_interface(ques," ","निष्प्रवाणिश्च
5.4.160");
    }
    ^nir[ ]pravANI{sup}[ ]bahuvrIhi {
        samAsAnwa_cgi_interface(ques," ","निष्प्रवाणिश्च
5.4.160");
    }
    /* 5.4.156 taddhita pratyaya */
    BrAwq{sup}[ ]({word}|su)[ ]bahuvrIhi {
        sprintf(ques,"किमिदं %s पुजार्थे वर्तते(Y/N)?",
p2_utf);
        samAsAnwa_cgi_interface(ques," ","वन्दिते भ्रातुः
5.4.157");
    }
    ^({word}|su)[ ]BrAwq{sup}[ ]bahuvrIhi {
        sprintf(ques,"किमिदं %s पुजार्थे वर्तते(Y/N)?",
p2_utf);
        samAsAnwa_cgi_interface(ques," ","वन्दिते भ्रातुः
5.4.157");
    }
    /* 5.4.158 -- Canxasi */
    ^(nAdI|wanwrI){sup}[ ]{word}[ ]bahuvrIhi {
        sprintf(ques,"अनेन समासेन स्वाङ्गः
उच्यते(Y/N)?");
        samAsAnwa_cgi_interface(ques,"
","नाडीतन्त्र्योः स्वाङ्गे 5.4.159");
    }
    ^{word}[ ](nAdI|wanwrI){sup}[ ]bahuvrIhi {
        sprintf(ques,"अनेन समासेन स्वाङ्गः
उच्यते(Y/N)?");
        samAsAnwa_cgi_interface(ques,"
","नाडीतन्त्र्योः स्वाङ्गे 5.4.159");
    }
    ^{prA}[iIuUq]{sup}[ ]{word}[ ]bahuvrIhi {
        sprintf(ques,"किमिदं %s
नद्यन्तं(Y/N)?",p2_utf);
        samAsAnwa_cgi_interface(ques,"kap","नदृतश्च 5.4.153");
    }
    ^{word}[ ]{prA}[iIuUq]{sup}[ ]bahuvrIhi {
        sprintf(ques,"किमिदं %s
नद्यन्तं(Y/N)?",p2_utf);

```

```

samAsAnwa_cgi_interface(ques,"kap","नदृतश्च 5.4.153");
    }
^{\word}{ }{\word}{ }bahuvrIhi
    {
    sprintf(ques,"अनेन समासेन कर्मव्यतिहारः
उच्यते(Y/N)?");
    samAsAnwa_cgi_interface(ques,"ic","इच्
कर्मव्यतिहारे 5.4.127");
    //sprintf(ques,"अनेन समासेन संज्ञा
उच्यते(Y/N)?");
    //samAsAnwa_cgi_interface(ques," "," न
संज्ञायां 5.4.155"); /* Amba */
    //sprintf(ques,"किम् अनेन समासेन शेषः
उच्यते(Y/N)?");
    /* SeRa? */

//samAsAnwa_cgi_interface(ques,"kap","शेषाद् विभाषा 5.4.154");
    }
^.* { samAsAnwa_cgi_interface("NULL","", ""); }
/* 1-4-61 UryAxicvidAcaH ca */
/* 1-4-74 sAkRAwpraBqwIni ca */

%%
char ques[LARGE];
char p1_utf[MEDIUMUTF];
char p2_utf[MEDIUMUTF];
char p1[MEDIUM];
char p2[MEDIUM];
char sup1[MEDIUM];
char sup2[MEDIUM];
char samAsaprakAra[MEDIUM];
char viXayakasUwra[MEDIUMUTF];
char avigraha[MEDIUMUTF];
char encoding[MEDIUM];

main (int argc, char *argv[]) {
char input[MEDIUM];

ques[0] = '\0';

strcpy(encoding,argv[1]);
strcpy(avigraha,argv[2]);
strcpy(p1,argv[3]);
strcpy(sup1,argv[4]);

```

```

strcpy(p2,argv[5]);
strcpy(sup2,argv[6]);
strcpy(viXAyakasUwra,argv[7]);
strcpy(samAsaparakAra,argv[8]);

strcpy(input,avigraha);
strcat(input," ");
strcat(input,samAsaparakAra);

cnvrtwx2utf(p1,p1_utf);
cnvrtwx2utf(p2,p2_utf);

call_samAsAnwalex(input);
}

call_samAsAnwalex(char *in){
    yy_scan_string(in);
    yylex();
}

samAsAnwawrap()
{
}

samAsAnwa_cgi_interface(char *question, char *prawyaya, char *sUwra)
{
    char question_utfr[MEDIUMUTF];
    if(!strcmp(question,"NULL")){
        printf("<td><form name=\"f2\" method=\"post\">\n");
    } else {
        printf("</tr><tr><td colspan=\"3\"><form name=\"f2\"
method=\"post\">\n");
    }
    printf("<input type=\"hidden\" name=\"encodingsamAsAnwa\"
id=\"encodingsamAsAnwa\" value=\"%s\" />",encoding);
    printf("<input type=\"hidden\" name=\"vighrasamAsAnwa\"
id=\"vighrasamAsAnwa\" value=\"%s\" />",avigraha);
    printf("<input type=\"hidden\" name=\"p1samAsAnwa\"
id=\"p1samAsAnwa\" value=\"%s\" />",p1);
    printf("<input type=\"hidden\" name=\"p2samAsAnwa\"
id=\"p2samAsAnwa\" value=\"%s\" />",p2);
    printf("<input type=\"hidden\" name=\"s1samAsAnwa\"
id=\"s1samAsAnwa\" value=\"%s\" />",sup1);

```

```

printf("<input type=\"hidden\" name=\"s2samAsAnwa\"
id=\"s2samAsAnwa\" value=\"%s\" />",sup2);
printf("<input type=\"hidden\" name=\"viXAyakasUwrasamAsAnwa\"
id=\"viXAyakasUwrasamAsAnwa\" value=\"%s\" />",viXAyakasUwra);
printf("<input type=\"hidden\" name=\"samAsaparakArasamAsAnwa\"
id=\"samAsaparakArasamAsAnwa\" value=\"%s\" />",samAsaparakAra);
printf("<input type=\"hidden\" name=\"samAsAnwa\" id=\"samAsAnwa\"
value=\"%s\" />",prawyaya);
printf("<input type=\"hidden\" name=\"sUwrasamAsAnwa\"
id=\"sUwrasamAsAnwa\" value=\"%s\" />",sUwra);
if(!strcmp(question,"NULL")){
printf("<input type=\"submit\" name=\"anssamAsAnwa\"
value=\"Continue\" id=\"anssamAsAnwa\" onclick=\"return
samAsAnwaNullcgi()\" />");
} else {
if(!strcmp(encoding,"RMN")) {
cnvrtutfd2r(question,question_utfr);
printf("%s",question_utfr);
} else {
printf("%s",question);
}
printf("<input type=\"submit\" name=\"anssamAsAnwa\" value=\"Yes\"
id=\"anssamAsAnwa\" onclick=\"return samAsAnwayescgi()\" />");
printf("<input type=\"submit\" name=\"anssamAsAnwa\" value=\"No\"
id=\"anssamAsAnwa\" onclick=\"return samAsAnwanocgi()\" />");
}
printf("</form></td></tr></table>");
}

```

Upasarjana.lex

```
%{
#include "struct.h"
extern char ques[LARGE];
extern char p1_utf[MEDIUMUTF];
extern char p2_utf[MEDIUMUTF];
}%

prA  [a-zA-Z]+

praWamA  (su|O|jas)
xviwIyA  (am|Ot|Sas)
wqwIyA  (tA|ByAm|Bis)
cawurWI  (fe|ByAm|Byas)
paFcamI  (fasi|ByAm|Byas)
RaRTI    (fas|os6|Am)
sapwamI  (fi|os7|sup)

sup  (su|O|jas|am|Ot|Sas|tA|ByAm3|Bis|fe|ByAm4|Byas4|fasi|ByAm5|Byas5|
fas|os6|Am|fi|os7|sup)

word  {prA}\+{sup}

SONdAxi  (SONda|XUrwa|kiwava|vyAda|pravINa|saMvIwa|anwara|aXi|patu|
paNdita|kuRala|capala|nipuNa|surARtra|baxXa)
safKyA    (eka|xvi|wri|cawvAri|paFca|RaRTa|sapwa|aRTa|nava|xaSa|
ekAxaSa|xvAxaSa|wrayoxaSa|cawurxaSa|paFcaxaSa|RodaSa|sapwaxaSa|
aRTAxaSa|navaxaSa|viMSawi|ekaviMSawi|xvAviMSawi|wrayoviMSawi|
cawurviMSaWi|paFcaviMSawi|RadviMSaWi|sapwaviMSawi|aRtAviMSawi|
navaviMSawi|wriMSaw|ekawriMSaw|xvAwriMSaw|wrayaswriMSaw|
cawuswriMSaw|paFcawriMSaw|RatwriMSaw|sapwawriMSaw|aRTAwriMSaw|
navawriMSaw|cawvAriMSaw|ekacawvAriMSaw|xvicawvAriMSaw|
wricawvAriMSaw|cawuscawvAriMSaw|paFcacawvAriMSaw|
RatcawvAriMSaw|sapwacawvAriMSaw|aRtAcawvAriMSaw|
navacawvAriMSaw|paFcASaW)
avy216    (upa|aXi|anu|nir|apa|xur|awi|prawi|pari|yAvax|A|saha|anwar)
xik  (pUrva|apara|uwwara|xakRiNa)
praSaMsA  (mawallikA|macarcikA|prakANdam|uxXa|wallaja)
yAjakAxi  (yAjaka|pUjaka|paricAraka|pariveRaka|pariRecaka|snApaka|
aXyApaka|uwsAha|uxvarwaka|howq|Bawq|raWagaNaka|pawwigaNaka)
vyAGrAxi  (vyAGra|siMha|qkRa|qRaBa|canxana|vqka|vqRa|varAha|haswin|
waru|kuFjara|ruru|pqRaw|puNdarika|palASa|kiwava)
```

%%

```
^{\avy216}[ ]{\word}[ ](2\1\6\){upasarjana_cgi_interface("NULL",1);}
```

```
^{\word}[ ]{\avy216}[ ](2\1\6\){upasarjana_cgi_interface("NULL",2);}
```

```
^yaWA[ ]{\word}[ ](2\1\7\){
    sprintf(ques,"यथा इति अव्ययम् असादृश्यार्थे
अस्ति(Y/N)?");
    upasarjana_cgi_interface(ques,1);
}
```

```
^{\word}[ ]yaWA[ ](2\1\7\){
    sprintf(ques,"यथा इति अव्ययम् असादृश्यार्थे
अस्ति(Y/N)?");
    upasarjana_cgi_interface(ques,2);
}
```

```
^yAvaw[ ]{\word}[ ](2\1\8\){
    sprintf(ques,"यावत् इति अव्ययम् अवधारणार्थे
अस्ति(Y/N)?");
    upasarjana_cgi_interface(ques,1);
}
```

```
^{\word}[ ]yAvaw[ ](2\1\8\){
    sprintf(ques,"यावत् इति अव्ययम् अवधारणार्थे
अस्ति(Y/N)?");
    upasarjana_cgi_interface(ques,2);
}
```

```
^{\word}[ ]prawi[ ](2\1\9\){
    sprintf(ques,"प्रति इति अव्ययम् मात्रार्थे अस्ति(Y/N)?");
    upasarjana_cgi_interface(ques,1);
}
```

```
^prawi[ ]{\word}[ ](2\1\9\){
    sprintf(ques,"यावत् इति अव्ययम् अवधारणार्थे
अस्ति(Y/N)?");
    upasarjana_cgi_interface(ques,2);
}
```

```
^(akRa|SalAkA|{safKyA})\+{wqwIyA}[ ](2\1\10\)  
{ upasarjana_cgi_interface("NULL",1);}
```

```
^pari[ ](akRa|SalAkA|{safKyA})\+{wqwIyA}[ ](2\1\10\)  
{ upasarjana_cgi_interface("NULL",2);}
```

```
^(apa|pari|bahis)[ ]{prA}\+{paFcamI}[ ](2\1\12\)  
{ upasarjana_cgi_interface("NULL",1);}
```

```
^{prA}\+{paFcamI}[ ](apa|pari|bahis)[ ](2\1\12\)  
{ upasarjana_cgi_interface("NULL",2);}
```

```
^A[ ]{word}[ ](2\1\13\){  
    sprintf(ques,"आङ् इति अव्ययम् मार्यादा एवम् अभिविधि अर्थे  
वर्तते(Y/N)?");  
    upasarjana_cgi_interface(ques,1);  
}
```

```
^{word}[ ]A[ ](2\1\13\){  
    sprintf(ques,"आङ् इति अव्ययम् मार्यादा एवम् अभिविधि अर्थे  
वर्तते(Y/N)?");  
    upasarjana_cgi_interface(ques,2);  
}
```

```
^(aBi|prawi)[ ]{word}[ ](2\1\14\){  
    sprintf(ques,"अभिप्रती आभिमुख्यार्थे वर्तते एवम् इदम्  
पदम् %s चिह्नवाचकम् वा(Y/N)?" ,p2_utf);  
    upasarjana_cgi_interface(ques,1);  
}
```

```
^{word}[ ](aBi|prawi)[ ](2\1\14\){  
    sprintf(ques,"अभिप्रती आभिमुख्यार्थे वर्तते एवम् इदम्  
पदम् %s चिह्नवाचकम् वा(Y/N)?" ,p1_utf);  
    upasarjana_cgi_interface(ques,2);  
}
```

```
^anu[ ]{word}[ ](2\1\15\){  
    sprintf(ques,"किम् अनु इति अव्ययम् समयार्थे वर्तते एवम्  
इदम् पदम् %s चिह्नवाचकम् वा(Y/N)?" ,p2_utf);  
    upasarjana_cgi_interface(ques,1);  
}
```

```

^{\word}[ ]anu[ ](2\1\15\)\ {
    sprintf(ques,"किम् अनु इति अव्ययम् समयार्थे वर्तते एवम्
इदम् पदम् %s इस् चिह्नवाचकम् वा(Y/N)?",p1_utf);
    upasarjana_cgi_interface(ques,2);
}
^anu[ ]{\word}[ ](2\1\16\)\ {
    sprintf(ques,"अनुना दैर्घ्य उच्यते(Y/N)?");
    upasarjana_cgi_interface(ques,1);
}
^{\word}[ ]anu[ ](2\1\16\)\ {
    sprintf(ques,"अनुना दैर्घ्य उच्यते(Y/N)?");
    upasarjana_cgi_interface(ques,2);
}
^{pAre|mAXye)[ ]{\prA}\+{\RaRTI}[ ](2\1\18\)\
    { upasarjana_cgi_interface("NULL",1);}

^{prA}\+{\RaRTI}[ ](pAre|mAXye)[ ](2\1\18\)\
    { upasarjana_cgi_interface("NULL",2);}

^{(\safKyA)}\+{\xviwIyA}[ ]{\word}[ ](2\1\19\)\ {
    sprintf(ques,"%s इति पदम् वंश्य वाचकं
वा(Y/N)?",p2_utf);
    upasarjana_cgi_interface(ques,1);
}

^{\word}[ ](\safKyA)}\+{\xviwIyA}[ ](2\1\19\)\ {
    sprintf(ques,"%s इति पदम् वंश्य वाचकं
वा(Y/N)?",p1_utf);
    upasarjana_cgi_interface(ques,2);
}
^{(\safKyA)}\+{\RaRTI}[ ](yamunA|goxAvarI|kqRNA|gafgA)\+{\sup}[ ]\
(2\1\20\)\ { upasarjana_cgi_interface("NULL",1);}

^{(yamunA|goxAvarI|kqRNA|gafgA)}\+{\sup}[ ](\safKyA)}\+{\RaRTI}[ ]\
(2\1\20\)\ { upasarjana_cgi_interface("NULL",2);}

^{\word}[ ](yamunA|gafgA|goxAvarI)\+{\praWamA}[ ](2\1\21\)\ {
    sprintf(ques,"अनेन समासेन
सञ्ज्ञा एवम् अन्यपदार्थः उच्यते(Y/N)?");
    upasarjana_cgi_interface(ques,1);
}

```

```

^(yamunA|gafgA|goxAvarI)\+{praWamA}[ ]{word}[ ](2\1\21\)\
                                                                    {
                                                                    sprintf(ques,"अनेन समासेन
सञ्ज्ञा एवम् अन्यपदार्थः उच्यते(Y/N)?");

upasarjana_cgi_interface(ques,2);

                                                                    }

^{prA}\+{xviwIyA}[ ](Sriwa|awIwa|pawiwa|gawa|awyaswa|prApwa|Apanna)\
+{sup}[ ](2\1\24\)\
    { upasarjana_cgi_interface("NULL",1);}

^(Sriwa|awIwa|pawiwa|gawa|awyaswa|prApwa|Apanna)\+{sup}[ ]{prA}\+
{xviwIyA}[ ](2\1\24\)\
    { upasarjana_cgi_interface("NULL",2);}

^svayam[ ]{word}[ ](2\1\25\){ if(chk_kqw(p2_utf,"kwa"))
{upasarjana_cgi_interface("NULL",1);} else
{upasarjana_cgi_interface("",0);} }

^{word}[ ]svayam[ ](2\1\25\){ if(chk_kqw(p1_utf,"kwa"))
{ upasarjana_cgi_interface("NULL",2); }else {upasarjana_cgi_interface("",0);}
}

^KatvA\+{xviwIyA}[ ]{word}[ ](2\1\26\)\
{ if(chk_kqw(p2_utf,"kwa"))
{upasarjana_cgi_interface("NULL",1);} else
{upasarjana_cgi_interface("",0);} }

^{word}[ ]KatvA\+{xviwIyA}[ ](2\1\26\)\
{ if(chk_kqw(p1_utf,"kwa"))
{ upasarjana_cgi_interface("NULL",2); } else
{upasarjana_cgi_interface("",0);} }

^sAmi[ ]{word}[ ](2\1\27\)\
    { if(chk_kqw(p2_utf,"kwa"))
{upasarjana_cgi_interface("NULL",1);} else
{upasarjana_cgi_interface("",0);} }

^{word}[ ]sAmi[ ](2\1\27\)\
    { if(chk_kqw(p1_utf,"kwa"))
{ upasarjana_cgi_interface("NULL",2); }else
{upasarjana_cgi_interface("",0);} }

^{prA}\+{xviwIyA}[ ](pramiwa|mukwa|pawiwa|kqwa)\+{sup}[ ](2\1\28\)\
    {upasarjana_cgi_interface("NULL",1);}

^(pramiwa|mukwa|pawiwa|kqwa)\+{sup}[ ]{prA}\+{xviwIyA}[ ](2\1\28\)\
    {upasarjana_cgi_interface("NULL",2);}

```

```

^{\prA}\+{\xviwIyA}[ ]{\word}[ ](2\1\29\ {
    sprintf(ques,"अनेन समासेन अत्यन्तसंयोगः
उच्यते(Y/N)?");
    upasarjana_cgi_interface(ques,1);
}

^{\word}[ ]{\prA}\+{\xviwIyA}[ ](2\1\29\ {
    sprintf(ques,"अनेन समासेन अत्यन्तसंयोगः
उच्यते(Y/N)?");
    upasarjana_cgi_interface(ques,2);
}

^{\prA}\+{\wqwIyA}[ ](manWa|arWa|paFcama)\+{\praWamA}[ ](2\1\30\
{upasarjana_cgi_interface("NULL",1);}

^(manWa|arWa)\+{\praWamA}[ ]{\prA}\+{\wqwIyA}[ ](2\1\30\
{upasarjana_cgi_interface("NULL",2);}

^{\prA}\+{\wqwIyA}[ ]{\word}[ ](2\1\30\ {
    sprintf(ques,"किम् अनेन समासेन तत्कृतार्थः
उच्यते(Y/N)?");
    upasarjana_cgi_interface(ques,1);
}

^{\word}[ ]{\prA}\+{\wqwIyA}[ ](2\1\30\ {
    sprintf(ques,"किम् अनेन समासेन तत्कृतार्थः
उच्यते(Y/N)?");
    upasarjana_cgi_interface(ques,2);
}

^{\prA}\+{\wqwIyA}[ ](pUrva|saxqSa|sama|Una|vikala|kalaha|nipuNa|miSra|
SlakRNa)+{\sup}[ ](2\1\31\ {upasarjana_cgi_interface("NULL",1);}

^(pUrva|saxqSa|sama|Una|vikala|kalaha|nipuNa|miSra|SlakRNa)\+{\sup}[ ]
{\prA}\+{\wqwIyA}[ ](2\1\31\ {upasarjana_cgi_interface("NULL",2);}

^{\prA}\+{\wqwIyA}[ ]{\word}[ ](2\1\32\ { if(chk_kqw(p2_utf,"kwa"))
{upasarjana_cgi_interface("NULL",1);} else
{upasarjana_cgi_interface("",0);} }

^{\word}[ ]{\prA}\+{\wqwIyA}[ ](2\1\32\ { if(chk_kqw(p1_utf,"kwa"))
{ upasarjana_cgi_interface("NULL",2); } else
{upasarjana_cgi_interface("",0);} }

```

/* 2.1.33 kqwya_pratyaya */

```
^{\prA}\+{\wqwIyA}[ ](anna|oxana)\+{\praWamA}[ ](2\1\34\)\
    sprintf(ques,"%s इति पदं व्यञ्जनवाचकम् एवम् %s
इति पदम् अन्नवाचकं वा(Y/N)?",p1_utf,p2_utf);
    upasarjana_cgi_interface(ques,1);
    }
```

```
^(anna|oxana)\+{\praWamA}[ ]{\prA}\+{\wqwIyA}[ ](2\1\34\)\
    sprintf(ques,"%s इति पदं व्यञ्जनवाचकम् एवम्
%s इति पदम् अन्नवाचकं वा(Y/N)?",p2_utf,p1_utf);
    upasarjana_cgi_interface(ques,2);
    }
```

```
^{\prA}\+{\wqwIyA}[ ]{\prA}\+{\praWamA}[ ](2\1\35\)\
    sprintf(ques,"%s इति पदं मिश्रीकरणवाचकं एवं %s
इति पदम् भक्ष्यवाचकं वा(Y/N)?",p1_utf,p2_utf);
    upasarjana_cgi_interface(ques,1);
    }
```

```
^{\prA}\+{\praWamA}[ ]{\prA}\+{\wqwIyA}[ ](2\1\35\)\
    sprintf(ques,"%s इति पदं
मिश्रीखरणवाचकं एवं %s इति पदम् भक्ष्यवाचकं वा(Y/N)?",p2_utf,p1_utf);
    upasarjana_cgi_interface(ques,2);
    }
```

```
^{\prA}\+{\cawurWI}[ ](bali|hiwa|suKa|rakRiwa|arWa|paxa)\+{\praWamA}[ ]\
(2\1\36\)\ {upasarjana_cgi_interface("NULL",1);}
```

```
^(bali|hiwa|suKa|rakRiwa|arWa|paxa)\+{\praWamA}[ ]{\prA}\+{\cawurWI}[ ]\
(2\1\36\)\ {upasarjana_cgi_interface("NULL",2); }
```

```
^{\prA}\+{\cawurWI}[ ]{\prA}\+{\sup}[ ](2\1\36\)\
    sprintf(ques,"%s इदम् पदम् %s अस्य पदस्य तदर्थो
वा (Y/N)?",p1_utf,p2_utf);
    upasarjana_cgi_interface(ques,1);
    }
```

```
^{\prA}\+{\sup}[ ]{\prA}\+{\cawurWI}[ ](2\1\36\)\
    sprintf(ques,"%s इदम् पदम् %s अस्य पदस्य तदर्थो
वा (Y/N)?",p2_utf,p1_utf);
    upasarjana_cgi_interface(ques,2);
    }
```

```
^{\prA}\+{\paFcamI}[ ](Baya)\+{\sup}[ ](2\1\37\)
```

```

{upasarjana_cgi_interface("NULL",1);}

^(Baya)\+{sup}[ ]{prA}\+{paFcaml}[ ](2\1\37\
{upasarjana_cgi_interface("NULL",2);}

^{prA}\+{paFcaml}[ ](apewa|apoDa|mukwa|pawiwa|Apawraswa)\+
{praWamA}[ ](2\1\38\) {
    sprintf(ques,"%s इदम् पदम् अल्पपञ्चम्यं वर्तते वा
(Y/N)?",p1_utf);
    upasarjana_cgi_interface(ques,1);
}
^(apewa|apoDa|mukwa|pawiwa|Apawraswa)\+{praWamA}[ ]{prA}\+
{paFcaml}[ ](2\1\38\) {
    sprintf(ques,"%s इदम् पदम्
अल्पपञ्चम्यं वर्तते वा (Y/N)?",p1_utf);
    upasarjana_cgi_interface(ques,2);
}

^(swoka|anwika|xUra|aByASa|viprakqRta|kqcCra)\+{paFcaml}[ ]{word}[ ]\
(2\1\39\) { if(chk_kqw(p2_utf,"kwa"))
{upasarjana_cgi_interface("NULL",1);} else
{upasarjana_cgi_interface("",0);} }

^{word}[ ](swoka|anwika|xUra|aByASa|viprakqRta|kqcCra)\+{paFcaml}[ ]\
(2\1\39\) { if(chk_kqw(p1_utf,"kwa"))
{ upasarjana_cgi_interface("NULL",2); } else
{upasarjana_cgi_interface("",0);} }

^{prA}\+{sapwamI}[ ]{SONdAxi}\+{sup}[ ](2\1\40\
{upasarjana_cgi_interface("NULL",1);}

^{SONdAxi}\+{sup}[ ]{prA}\+{sapwamI}[ ](2\1\40\
{upasarjana_cgi_interface("NULL",2);}

^{prA}\+{sapwamI}[ ](sixXa|SuRka|pakva|banXa)\+{sup}[ ](2\1\41\
{upasarjana_cgi_interface("NULL",1);}

^(sixXa|SuRka|pakva|banXa)\+{sup}[ ]{prA}\+{sapwamI}[ ](2\1\41\
{upasarjana_cgi_interface("NULL",2);}

^{prA}\+{sapwamI}[ ](XvAfKRa)\+{praWamA}[ ](2\1\42\) {
    sprintf(ques,"अनेन समासेन क्षेपार्थः
उच्यते(Y/N)?");

```

```

        upasarjana_cgi_interface(ques,1);
    }

    ^{XvAfKRa}\+{praWamA}[ ]{prA}\+{sapwamI}[ ](2\1\42\) {
        sprintf(ques,"अनेन समासेन क्षेपार्थः
उच्यते(Y/N)?");

        upasarjana_cgi_interface(ques,2);
    }

    /* 2.1.43 kqwyā_pratyaya */

    ^{prA}\+{sapwamI}[ ]{word}[ ](2\1\43\) {
        sprintf(ques,"अनेन समासेन आवश्यकार्थः
उच्यते(Y/N)?");

        upasarjana_cgi_interface(ques,1);
    }

    ^{word}[ ]{prA}\+{sapwamI}[ ](2\1\43\) {
        sprintf(ques,"अनेन समासेन आवश्यकार्थः
उच्यते(Y/N)?");

        upasarjana_cgi_interface(ques,2);
    }

    ^{word}[ ]{prA}\+{sapwamI}[ ](2\1\44\) {
        sprintf(ques,"अनेन समासेन सञ्ज्ञा
उच्यते(Y/N)?");

        upasarjana_cgi_interface(ques,2);
    }

    ^{prA}\+{sapwamI}[ ]{word}[ ](2\1\44\) {
        sprintf(ques,"अनेन समासेन सञ्ज्ञा
गम्यते(Y/N)?");

        upasarjana_cgi_interface(ques,1);
    }

    ^{prA}\+{sapwamI}[ ]{word}[ ](2\1\45\) {
        if (chk_kqw(p2_utf,"kwa")) {
            sprintf(ques,"किम् %s इति पदम् अहोरात्रवयवः
वा(Y/N)?",p1_utf);

            upasarjana_cgi_interface(ques,1);
        } else { upasarjana_cgi_interface("NULL",0);}
    }

    ^{word}[ ]{prA}\+{sapwamI}[ ](2\1\45\) {

```

```

        if (chk_kqw(p1_utf,"kwa")) {
            sprintf(ques,"किम् %s इति पदम्
अहोरात्रवयवः वा(Y/N)?",p2_utf);
            upasarjana_cgi_interface(ques,2);
        } else { upasarjana_cgi_interface("NULL",0);}
    }

^wawra[ ]{word}[ ](2\1\46\){ if(chk_kqw(p2_utf,"kwa"))
{upasarjana_cgi_interface("NULL",1);} else
{upasarjana_cgi_interface("",0);} }

^{word}[ ]wawra[ ](2\1\46\){ if(chk_kqw(p1_utf,"kwa"))
{ upasarjana_cgi_interface("NULL",2); } else
{upasarjana_cgi_interface("",0);} }

^{prA}\+{sapwamI}[ ]{word}[ ](2\1\47\){
    sprintf(ques,"अनेन समासेन क्षेपार्थः
उच्यते(Y/N)?");
    upasarjana_cgi_interface(ques,1);
}

^{word}[ ]{prA}\+{sapwamI}[ ](2\1\47\){
    sprintf(ques,"अनेन समासेन क्षेपार्थः
उच्यते(Y/N)?");
    upasarjana_cgi_interface(ques,2);
}

/* 2.1.48 GaNapATha */

^(eka|sarva|jaraw|purANa|nava|kevala)\+{praWamA}[ ]{word}[ ](2\1\49\){
    {upasarjana_cgi_interface("NULL",1);}

^{word}[ ](eka|sarva|jaraw|purANa|nava|kevala)\+{praWamA}[ ](2\1\49\){
    {upasarjana_cgi_interface("NULL",2);}
/* 2.1.49 pUrvakAla */

^({safKyA}|{xik})\+{praWamA}[ ]{word}[ ](2\1\50\){
    sprintf(ques,"अनेन समासेन सञ्ज्ञा
गम्यते(Y/N)?");
    upasarjana_cgi_interface(ques,1);
}

^{word}[ ]({safKyA}|{xik})\+{praWamA}[ ](2\1\50\){
    sprintf(ques,"अनेन समासेन सञ्ज्ञा

```

```

गम्यते(Y/N)?");
                                upasarjana_cgi_interface(ques,2);
                                }
/* 2.1.51 waxXiwArWowwarapaxasamAhAre ca */

^{\safKyA}\+{\RaRTI}[ ]{\word}[ ](2\1\52\){
                                sprintf(ques,"अनेन समासेन द्विगुः अथवा
समाहारः उच्यते(Y/N)?");
                                upasarjana_cgi_interface(ques,1);
                                }

^{\word}[ ]{\safKyA}\+{\RaRTI}[ ](2\1\52\){
                                sprintf(ques,"अनेन समासेन द्विगुः अथवा
समाहारः उच्यते(Y/N)?");
                                upasarjana_cgi_interface(ques,2);
                                }

^{\word}[ ]{\word}[ ](2\1\53\){
                                sprintf(ques,"%s इति पदम् %s इति पदस्य निन्दार्थः
वा(Y/N)?",p1_utf,p2_utf);
                                upasarjana_cgi_interface(ques,2);
                                }
/* We have to ask reverse question as well */

^(pApa|aNaka)\+{\praWamA}[ ]{\word}[ ](2\1\54\){
                                sprintf(ques,"किम् %s इति पदम् निन्दार्थे
वर्तते(Y/N)?",p1_utf);
                                upasarjana_cgi_interface(ques,1);
                                }

^{\word}[ ](pApa|aNaka)\+{\praWamA}[ ](2\1\54\){
                                sprintf(ques,"किम् %s इति पदम्
निन्दार्थे वर्तते(Y/N)?",p2_utf);
                                upasarjana_cgi_interface(ques,2);
                                }

^{\word}[ ]{\word}[ ](2\1\55\){
                                sprintf(ques,"किं %s पदम् %s पदस्य उपमानं
वा(Y/N)?",p2_utf,p1_utf);
                                upasarjana_cgi_interface(ques,2);
                                }
/* We have to ask reverse question as well */

```

```

^{\word}[ ]{\vyAGrAxi}[ ](2\1\56\) {
    sprintf(ques,"किम् %s पदम् %s पदस्य उपमेयः
वा(Y/N)?",p1_utf,p2_utf);
    upasarjana_cgi_interface(ques,1);
}
^{\vyAGrAxi}[ ]{\word}[ ](2\1\56\) {
    sprintf(ques,"किम् %s पदम् %s पदस्य उपमेयः
वा(Y/N)?",p2_utf,p1_utf);
    upasarjana_cgi_interface(ques,2);
}

^{\word}[ ]{\word}[ ](2\1\57\){
    sprintf(ques,"किं %s पदं %s पदस्य विशेषणं वा(Y/N)?",
p1_utf,p2_utf);
    upasarjana_cgi_interface(ques,1);
}
/* We have to ask reverse question as well */

^(pUrva|apara|praWama|carama|jaGanya|samAna|maXya|maXyama|vIra)\+
{\praWamA}[ ]{\word}[ ](2\1\58\){upasarjana_cgi_interface("NULL",1);}

^{\word}[ ](pUrva|apara|praWama|carama|jaGanya|samAna|maXya|maXyama|
vIra)\+{\praWamA}[ ](2\1\58\){upasarjana_cgi_interface("NULL",2);}

/* 2.1.59 SreNyAxayaH kqwAxiBiH */
/* 2.1.60 kwena naFviSiRtenAnaF */

^(saw|mahaw|mahawI|parama|uwwama|uwkqRta)\+{\sup}[ ]{\word}[ ](
2\1\61\){

sprintf(ques,"अनेन समासेन पूज्यमानार्थः गम्यते(Y/N)?");

upasarjana_cgi_interface(ques,1);

}

^{\word}[ ](saw|mahaw|mahawI|parama|uwwama|uwkqRta)\+{\sup}[ ](
2\1\61\){

sprintf(ques,"अनेन समासेन पूज्यमानार्थः गम्यते(Y/N)?");

upasarjana_cgi_interface(ques,2);

}

```

```

^{\word}[ ](vqnxAraka|nAga|kuFjara)\+{\praWamA}[ ](2\1\62\)\ {
                                                                    sprintf(ques,"अनेन समासेन
पूज्यमानार्थः गम्यते(Y/N)?");

upasarjana_cgi_interface(ques,1);
                                                                    }

^(vqnxAraka|nAga|kuFjara)\+{\praWamA}[ ]{\word}[ ](2\1\62\)\ {
                                                                    sprintf(ques,"अनेन समासेन
पूज्यमानार्थः गम्यते(Y/N)?");

upasarjana_cgi_interface(ques,2);
                                                                    }

^(kawara|kawama)\+{\praWamA}[ ]{\word}[ ](2\1\63\)\ {
                                                                    sprintf(ques,"किम् %s इति पदम्
जातिपरिप्रश्नार्थ सूचयति(Y/N)?",p1_utf);
                                                                    upasarjana_cgi_interface(ques,1);
                                                                    }

^{\word}[ ](kawara|kawama)\+{\praWamA}[ ](2\1\63\)\ {
                                                                    sprintf(ques,"किम् %s इति पदम्
जातिपरिप्रश्नार्थ सूचयति(Y/N)?",p2_utf);
                                                                    upasarjana_cgi_interface(ques,2);
                                                                    }

^kim[ ]{\word}[ ](2\1\64\)\ {
                                                                    sprintf(ques,"किम् इति अव्ययम् क्षेपार्थे वर्तते(Y/N)?");
                                                                    upasarjana_cgi_interface(ques,1);
                                                                    }

^{\word}[ ]kim[ ](2\1\64\)\ {
                                                                    sprintf(ques,"किम् इति अव्ययम् क्षेपार्थे वर्तते(Y/N)?");
                                                                    upasarjana_cgi_interface(ques,2);
                                                                    }

^{\prA}\+{\sup}[ ]{\prA}\+{\sup}[ ](2\1\60\)\ {
/* We assume that p1_utf and p2_utf are negations of each other. So we do
not allow kqwa=aBukwa kind of samAsas. p1_utf and p2_utf are assumed to
be kqws from the same XAwu, one with negation and other without */
                                                                    if( chk_kqw(p1_utf,"kwa") &&
                                                                    (!strcmp(p1_utf,p2_utf+1) && p2_utf[0] == 'a') ||
                                                                    (!strcmp(p1_utf,p2_utf+2) && (p2_utf[0] == 'a')

```

```

&& (p2_utf[1] == 'n'))
    ) { upasarjana_cgi_interface("NULL",1); }
    else if( chk_kqw(p2_utf,"kwa") &&
    ((!strcmp(p2_utf,p1_utf+1) && p1_utf[0] == 'a') ||
    (!strcmp(p2_utf,p1_utf+2) && (p1_utf[0] == 'a')
&& (p2_utf[1] == 'n'))
    ) { upasarjana_cgi_interface("NULL",2); }
    else {upasarjana_cgi_interface("NULL",0);}
}

^{word}[ ](potA|yuvawi|swoka|kawipas|gqRti|Xenu|vaSA|vehaw|baRkayaNI|
pravakkwq|Srowriya|aXyApaka|XUrWa)\+{praWamA}[ ](2\1\65\ ) {
    sprintf(ques,"किम् %s इति पदम् जातिवाचकः
वा(Y/N)?",p1_utf);
    upasarjana_cgi_interface(ques,1);
}

^(potA|yuvawi|swoka|kawipas|gqRti|Xenu|vaSA|vehaw|baRkayaNI|pravakkwq|
Srowriya|aXyApaka|XUrWa)\+{praWamA}[ ]{word}[ ](2\1\65\ ) {
    sprintf(ques,"किम् %s इति
पदम् जातिवाचकं वा(Y/N)?",p2_utf);
    upasarjana_cgi_interface(ques,2);
}

^{word}[ ]{praSaMsA}\+{praWamA}[ ](2\1\66\ ) {
    sprintf(ques,"किम् %s इति पदम् जातिवाचकं
वा(Y/N)?",p1_utf);
    upasarjana_cgi_interface(ques,1);
}

^{praSaMsA}\+{praWamA}[ ]{word}[ ](2\1\66\ ) {
    sprintf(ques,"किम् %s इति पदम् जातिवाचकः
वा(Y/N)?",p2_utf);
    upasarjana_cgi_interface(ques,2);
}

^yuvan\+{praWamA}[ ](KalawI|paliwa|valinaljarawI)\+{praWamA}[ ]\
(2\1\67\ ) {upasarjana_cgi_interface("NULL",1);}

^(KalawI|paliwa|valinaljarawI)\+{praWamA}[ ]yuvan\+{praWamA}[ ]\
(2\1\67\ ) {upasarjana_cgi_interface("NULL",2);}

```

/* 2.1.68 kqwyawulyAKyA ajAwyA */

```
^{\word}[ ]{\word}[ ](2\1\69\){
    sprintf(ques,"%s %s किम् उभावपि पदौ
वर्णवाचकौ(Y/N)?",p2_utf,p1_utf);
    upasarjana_cgi_interface(ques,1);
}
```

```
^kumArI\+{\praWamA}[ ](SramaNA|pravrajiwA|garBiNI)\+{\praWamA}[ ]\
(2\1\70\){ upasarjana_cgi_interface("NULL",1);}
```

```
^(SramaNA|pravrajiwA|garBiNI)\+{\praWamA}[ ]kumArI\+{\praWamA}[ ]\
(2\1\70\){ upasarjana_cgi_interface("NULL",2);}
```

```
^{\word}[ ]garBiNI\+{\praWamA}[ ](2\1\71\){
    sprintf(ques,"किम् %s इति पदम्
चतुष्पाज्जातिवाचको वा (Y/N)?",p1_utf);
    upasarjana_cgi_interface(ques,1);
}
```

```
^garBiNI\+{\praWamA}[ ]{\word}[ ](2\1\71\){
    sprintf(ques,"किम् %s इति पदम्
चतुष्पाज्जातिवाचको वा (Y/N)?",p2_utf);
    upasarjana_cgi_interface(ques,2);
}
```

```
^(pUrva|apara|aXara|uwwara)\+{\sup}[ ]{\word}[ ](2\2\1\){
    sprintf(ques,"किम् %s इति पदम्
%s पदस्य अवयवो वा (Y/N)?",p1_utf,p2_utf);
    upasarjana_cgi_interface(ques,1);
}
```

```
^{\word}[ ](pUrva|apara|aXara|uwwara)\+{\sup}[ ](2\2\1\){
    sprintf(ques,"किम् %s इति पदम्
%s पदस्य अवयवो वा(Y/N)?",p2_utf,p1_utf);
    upasarjana_cgi_interface(ques,2);
}
```

```
^arXa\+{\praWamA}[ ]{\word}[ ](2\2\2\){
    sprintf(ques,"किम् %s इति पदम् अवयवी वा %s इति
पदस्य (Y/N)?",p1_utf,p2_utf);
```

```

        upasarjana_cgi_interface(ques,1);
    }

^{\word}[ ]arXa\+{\praWamA}[ ](2\2\2\)\ {
    sprintf(ques,"किम् %s इति पदम् %s पदस्य अवयवो
वा (Y/N)?" ,p2_utf,p1_utf);
    upasarjana_cgi_interface(ques,2);
}

^(xviwIya|wqwIya|cawurWa|wurya)\+{\praWamA}[ ]{\word}[ ](2\2\3\)\ {
    sprintf(ques,"किम् %s इति पदं %s इति पदस्य
अवयवः वा (Y/N)?" ,p1_utf,p2_utf);
    upasarjana_cgi_interface(ques,1);
}

^{\word}[ ](xviwIya|wqwIya|cawurWa|wurya)\+{\praWamA}[ ](2\2\3\)\ {
    sprintf(ques,"किम् %s इति पदं %s इति पदस्य
अवयवः वा (Y/N)?" ,p2_utf,p1_utf);
    upasarjana_cgi_interface(ques,2);
}

^{\prA}\+{\xviwIyA}[ ](prApwa|Apanna)\+{\praWamA}[ ](2\2\4\)\
{upasarjana_cgi_interface("NULL",1);}

^(prApwa|Apanna)[ ]{\prA}\+{\xviwIyA}[ ](2\2\4\)\
{upasarjana_cgi_interface("NULL",2);}

^{\word}[ ]{\prA}\+{\RaRTI}[ ](2\2\5\)\ {
    sprintf(ques,"किम् %s इति पदं कालवाचकः एवम् %s
इति पदं परिच्छेद्यवाचकः वा(Y/N)?" ,p1_utf,p2_utf);
    upasarjana_cgi_interface(ques,1);
}

^{\prA}\+{\RaRTI}[ ]{\word}[ ](2\2\5\)\ {
    sprintf(ques,"किम् %s इति पदं कालवाचकः एवम् %s
इति पदं परिच्छेद्यवाचकः वा(Y/N)?" ,p2_utf,p1_utf);
    upasarjana_cgi_interface(ques,2);
}

^n[ ]{\word}[ ](2\2\6\)\ {upasarjana_cgi_interface("NULL",1);}

^{\word}[ ]n[ ](2\2\6\)\ {upasarjana_cgi_interface("NULL",2);}

```

```

^IRaw[ ]{word}[ ](2\2\7\){ if(!chk_kqw(p2_utf,"kwa"))
{upasarjana_cgi_interface("NULL",1);} else
{upasarjana_cgi_interface("",0);} }

{word}[ ]^IRaw[ ](2\2\7\){ if(!chk_kqw(p1_utf,"kwa"))
{upasarjana_cgi_interface("NULL",2);} else
{upasarjana_cgi_interface("",0);} }

^{\prA}\+{\RaRTI}[ ]{word}[ ](2\2\8\){
upasarjana_cgi_interface("NULL",1);
}

^{\word}[ ]{\prA}\+{\RaRTI}[ ](2\2\8\){
upasarjana_cgi_interface("NULL",2);
}

^{\prA}\+{\RaRTI}[ ]{\yAjakAxi}\+{\praWamA}[ ](2\2\9\){
upasarjana_cgi_interface("NULL",1);}

^{\yAjakAxi}\+{\praWamA}[ ]{\prA}\+{\RaRTI}[ ](2\2\9\){
upasarjana_cgi_interface("NULL",2);}

^{\word}[ ]{\prA}\+{\RaRTI}[ ](2\2\10\){
printf(ques,"किम् %s इति पदम् निर्धारणे
वर्तते(Y/N)?",p2_utf);
upasarjana_cgi_interface(ques,2);
}

^{\prA}\+{\RaRTI}[ ]{word}[ ](2\2\10\){
printf(ques,"किम् %s इति पदम् निर्धारणे
वर्तते(Y/N)?",p1_utf);
upasarjana_cgi_interface(ques,1);
}

/* 2.2.11
pUraNaguNasuhiwArWasaxavyayawavyasamAnAXikaraNena */
/* 2.2.12 kwenA ca pUjAyAm */
/* 2.2.13 aXikaraNavAcinA ca */

^{\word}[ ]{\prA}\+{\RaRTI}[ ](2\2\17\){
upasarjana_cgi_interface("NULL",2);}

^{\prA}\+{\RaRTI}[ ]{word}[ ](2\2\17\){

```

```

{upasarjana_cgi_interface("NULL",1);}

^{word}[ ](ku|pra|para|apa|sam|anu|ava|nis|nir|xus|xur|vi|A|ni|aXi|awi|api|su|ux|
aBi|prawi|para|upa)[ ](2\2\18\){upasarjana_cgi_interface("NULL",2);}

^(ku|pra|para|apa|sam|anu|ava|nis|nir|xus|xur|vi|A|ni|aXi|awi|api|su|ux|aBi|prawi|
para|upa)[ ]{word}[ ](2\2\18\){upasarjana_cgi_interface("NULL",1);}

^{word}[ ]{prA}\+{praWamA}[ ](2\2\18\){upasarjana_cgi_interface("NULL",2);}

^{prA}\+{praWamA}[ ]{word}[ ](2\2\18\){upasarjana_cgi_interface("NULL",1);}

/* 2.2.19: upapaxam awif */
^{prA}\+{xviwIyA}[ ]{word}[ ](2\2\19\){upasarjana_cgi_interface("NULL",2);}

^{word}[ ]({word}|avy)[ ](2\2\19\){upasarjana_cgi_interface("NULL",1);}

^{uccEH}[ ]{prA}[ ](2\2\22\){ if(chk_kqw(p2_utf,"kwvA"))
{upasarjana_cgi_interface("NULL",1);} else
{upasarjana_cgi_interface("",0);}}

^{prA}[ ]uccEH[ ](2\2\22\){ if(chk_kqw(p2_utf,"kwvA"))
{upasarjana_cgi_interface("NULL",1);} else
{upasarjana_cgi_interface("",0);}}

^{prA}\+{sapwamI}[ ]{word}[ ](3\1\92\){upasarjana_cgi_interface("NULL",1);}

^{word}[ ]{prA}\+{sapwamI}[ ](3\1\92\){upasarjana_cgi_interface("NULL",2);}

^(su|awi)[ ]{word}[ ](5\4\69\){upasarjana_cgi_interface("NULL",1);}

^{word}[ ](su|awi)[ ](5\4\69\){upasarjana_cgi_interface("NULL",2);}

^xvis[ ]wAvawI\+{praWamA}[ ](5\4\84\){upasarjana_cgi_interface("NULL",1);}

^wAvawI\+{praWamA}[ ]xvis[ ](5\4\84\){upasarjana_cgi_interface("NULL",1);}

```

```

        {upasarjana_cgi_interface("NULL",2);}

^.*                                {upasarjana_cgi_interface("NULL",1);}
%%
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

char ques[LARGE];

char p1_utf[MEDIUMUTF]; /* utf takes 3 times space */
char p2_utf[MEDIUMUTF]; /* utf takes 3 times space */

char encoding[MEDIUM];
char avigraha[MEDIUM];
char p1[MEDIUM];
char p2[MEDIUM];
char sup1[SMALL];
char sup2[SMALL];
char samAsaprakAra[MEDIUM];
char viXAyakasUwra[LARGE];
char samAsAnwa[MEDIUM];
char samAsaprakAra[MEDIUM];

void call_pUrvanipAwa_xvanxva();
void call_pUrvanipAwa_bahuvrIhi();

#define myPATH "/home/pavan/scl"

main (int argc, char *argv[]) {
int gawi;
int viBARA;
int upasarjana;
char sUwrastring[LARGE];
char input[MEDIUM];
char p1_utfr[MEDIUMUTF]; /* utf takes 3 times space */
char p2_utfr[MEDIUMUTF]; /* utf takes 3 times space */

    upasarjana = 0;
    strcpy(encoding,argv[1]);
    strcpy(avigraha,argv[2]);
    strcpy(p1,argv[3]);
    strcpy(sup1,argv[4]);

```

```

strcpy(p2,argv[5]);
strcpy(sup2,argv[6]);
strcpy(viXAyakasUwra,argv[7]);
strcpy(samAsaprakAra,argv[8]);
strcpy(samAsAnwa,argv[9]);

strcpy(input,avighraha);
strcat(input," (");
strcat(input,viXAyakasUwra);
strcat(input,")");

cnvrtwx2utf(p1,p1_utf);
cnvrtwx2utf(p2,p2_utf);

if(!strcmp(encoding,"RMN")){
    cnvrtwx2utf(p1_utf,p1_utfr);
    cnvrtwx2utf(p2_utf,p2_utfr);

    strcpy(p1_utfr, p1_utf);
    strcpy(p2_utfr, p2_utf);
}

upasarjana = 0;

if ( !strcmp(samAsaprakAra,"wawpuruRa") ||
    !strcmp(samAsaprakAra,"avyayIBAvH")
    ) {
    if (!strcmp(viXAyakasUwra,"2.2.18"))
        call_gawilex(avighraha);
    else call_upasarjanalex(input);
}
else if (!strcmp(samAsaprakAra,"bahuvrIhi"))
    call_pUrvanipAwa_bahuvrIhi (p1,sup1,p2,sup2);
else if (!strcmp(samAsaprakAra,"xvanxva"))
    call_pUrvanipAwa_xvanxva (p1,sup1,p2,sup2);
}

call_upasarjanalex(char *in){
    yy_scan_string(in);
    yylex();
}

yywrap()

```

```
{
}
```

```
upasarjana_cgi_interface(char *question,int position){
    char questionr[LARGE];

    if(strcmp(question,"NULL")){
        printf("</tr><td colspan=\"3\"><form name=\"f3\" method=\"post\" />");
    } else {
        printf("<td><form name=\"f3\" method=\"post\" />");
    }
    printf("<input type=\"hidden\" name=\"encodingupasarjana\"
id=\"encodingupasarjana\" value=\"%s\" />",encoding);
    printf("<input type=\"hidden\" name=\"vighraupasarjana\"
id=\"vighraupasarjana\" value=\"%s\" />",avighraha);
    printf("<input type=\"hidden\" name=\"p1upasarjana\" id=\"p1upasarjana\"
value=\"%s\" />",p1);
    printf("<input type=\"hidden\" name=\"p2upasarjana\" id=\"p2upasarjana\"
value=\"%s\" />",p2);
    printf("<input type=\"hidden\" name=\"s1upasarjana\" id=\"s1upasarjana\"
value=\"%s\" />",sup1);
    printf("<input type=\"hidden\" name=\"s2upasarjana\" id=\"s2upasarjana\"
value=\"%s\" />",sup2);
    printf("<input type=\"hidden\" name=\"samAsaparakAraupasarjana\"
id=\"samAsaparakAraupasarjana\" value=\"%s\" />",samAsaparakAra);
    //printf("in upasarjana lex samAsAnwau = %s",samAsAnwa);
    printf("<input type=\"hidden\" name=\"samAsAnwaupasarjana\"
id=\"samAsAnwaupasarjana\" value=\"%s\" />",samAsAnwa);
    printf("<input type=\"hidden\" name=\"viXAyakasUwraupasarjana\"
id=\"viXAyakasUwraupasarjana\" value=\"%s\" />",viXAyakasUwra);
    printf("<input type=\"hidden\" name=\"positionupasarjana\"
id=\"positionupasarjana\" value=\"%d\" />",position);
    printf("<input type=\"hidden\" name=\"dividupasarjana\"
id=\"dividupasarjana\" value=\"#output3\" />");
    if(strcmp(question,"NULL")){
        if(!strcmp(encoding,"RMN")) {
            cnvrtutfd2r(question,questionr);
            printf("%s",questionr);
        } else {
            printf("%s",question);
        }
    }
    printf("<input type=\"submit\" name=\"ansupasarjana\" value=\"Yes\"
id=\"ansupasarjana\" onclick=\"return upasarjanayescgi()\" />");
}
```

```

        printf("<input type=\"submit\" name=\"ansupasarjana\" value=\"No\"
id=\"ansupasarjana\" onclick=\"return upasarjananocgi()\" />");
    }
    else
        if((position == 1) || (position == 2))
            if (!strcmp(encoding,"RMN")) {
                printf("<input type=\"submit\" name=\"ansupasarjana\"
value=\"Continue\" id=\"ansupasarjana\" onclick=\"return
upasarjanaNullcgi()\" />");
            } else {
                printf("<input type=\"submit\" name=\"ansupasarjana\"
value=\"Continue\" id=\"ansupasarjana\" onclick=\"return
upasarjanaNullcgi()\" />");
            }
            else if (!strcmp(encoding,"RMN")) {
                printf("<input type=\"submit\" name=\"ansupasarjana\" value=\"
upasarjana saṃjñā nirdhāraṇa\" id=\"ansupasarjana\" onclick=\"return
upasarjanaNullcgi()\" />");
            } else {
                printf("<input type=\"submit\" name=\"ansupasarjana\"
value=\"उपसर्जन संज्ञा निर्धारण: Error in Data\" id=\"ansupasarjana\"
onclick=\"return upasarjanaNullcgi()\" />");
            }
            printf("</form></td></tr></table>");
    }
}

```

```

int chk_kqw(char *str, char *kqw_str){
FILE *fp;
char ans[SMALL];
char cmd[MEDIUM];
char fin[MEDIUM];
char fout[MEDIUM];
int pid;

```

```

pid = getpid();

```

```

sprintf(fin, "/tmp/tmp%d", pid);
sprintf(fout, "/tmp/result%d", pid);

```

```

fp = fopen(fin, "w");
fprintf(fp, "%s\n", str);
fclose(fp);

```

```

sprintf(cmd, "/usr/bin/lt-proc -c %s/morph_bin/skt_morf.bin < %s | grep -c
'kqw_prawyayaH:%s' > %s", myPATH, fin, kqw_str, fout);
system(cmd);

fp = fopen(fout, "r");
fscanf(fp, "%c", ans);
fclose(fp);

sprintf(cmd, "rm %s %s", fin, fout);
system(cmd);

if(ans[0] == '1') { return 1; } else { return 0; }
}

void call_pUrvanipAwa_bahuvrIhi (char *p1, char *s1, char *p2, char *s2) {
    char question[LARGE];
    char questionr[LARGE];
    int position;
    char sUwra[MEDIUMUTF];

    if(!strcmp(s1, "fi") || !strcmp(s1, "os") || !strcmp(s1, "sup")) {
        position = 1;
        strcpy(sUwra, "सप्तमी विशेषणे बहुव्रीहौ 2.2.35");
    } else if(!strcmp(s2, "fi") || !strcmp(s2, "os") || !strcmp(s2, "sup")) {
        position = 2;
    } else {
        sprintf(question, " किम् %s पदम् %s पदस्य विशेषणं वा(Y/N)" ,p1_utf,p2_utf);
        strcpy(sUwra, "सप्तमी विशेषणे बहुव्रीहौ 2.2.35");
        position = 0;
    }

    if(strcmp(question, "NULL")){
        printf("</tr><td colspan=\\\"3\\\"><form name=\\\"f3\\\" method=\\\"post\\\" />");
    } else {
        printf("<td><form name=\\\"f3\\\" method=\\\"post\\\" />");
    }
    printf("<input type=\\\"hidden\\\" name=\\\"encodingpUrvanipAwa\\\"
id=\\\"encodingpUrvanipAwa\\\" value=\\\"%s\\\" />", encoding);
    printf("<input type=\\\"hidden\\\" name=\\\"vigrahapUrvanipAwa\\\"
id=\\\"vigrahapUrvanipAwa\\\" value=\\\"%s\\\" />", avigraha);
    printf("<input type=\\\"hidden\\\" name=\\\"p1pUrvanipAwa\\\"
id=\\\"p1pUrvanipAwa\\\" value=\\\"%s\\\" />", p1);
    printf("<input type=\\\"hidden\\\" name=\\\"p2pUrvanipAwa\\\"

```

```

id="p2pUrvanipAwa\" value=\"%s\" />",p2);
    printf("<input type=\"hidden\" name=\"s1pUrvanipAwa\"
id=\"s1pUrvanipAwa\" value=\"%s\" />",sup1);
    printf("<input type=\"hidden\" name=\"s2pUrvanipAwa\"
id=\"s2pUrvanipAwa\" value=\"%s\" />",sup2);
    printf("<input type=\"hidden\" name=\"samAsaparakArapUrvanipAwa\"
id=\"samAsaparakArapUrvanipAwa\" value=\"%s\" />",samAsaparakAra);
    printf("<input type=\"hidden\" name=\"samAsAnwapUrvanipAwa\"
id=\"samAsAnwapUrvanipAwa\" value=\"%s\" />",samAsAnwa);
    printf("<input type=\"hidden\" name=\"viXAyakasUwra\"
id=\"viXAyakasUwra\" value=\"%s\" />",viXAyakasUwra);
    printf("<input type=\"hidden\" name=\"positionpUrvanipAwa\"
id=\"positionpUrvanipAwa\" value=\"%d\" />",position);
    printf("<input type=\"hidden\" name=\"sUwrapUrvanipAwa\"
id=\"sUwrapUrvanipAwa\" value=\"%s\" />",sUwra);
    printf("<input type=\"hidden\" name=\"dividpUrvanipAwa\"
id=\"dividpUrvanipAwa\" value=\"#output3\" />");
    if(strcmp(question,"NULL")){
        if(!strcmp(encoding,"RMN")) {
            cnvrtutfd2r(question,questionr);
            printf("%s",questionr);
        } else {
            printf("%s",question);
        }
        printf("<input type=\"submit\" name=\"anspUrvanipAwa\" value=\"Yes\"
id=\"anspUrvanipAwa\" onclick=\"return pUrvanipAwayscgi()\" />");
        printf("<input type=\"submit\" name=\"anspUrvanipAwa\" value=\"No\"
id=\"anspUrvanipAwa\" onclick=\"return pUrvanipAwanocgi()\" />");
    }
    else
        printf("<input type=\"submit\" name=\"anspUrvanipAwa\"
value=\"Continue\" id=\"anspUrvanipAwa\" onclick=\"return
pUrvanipAwaNullcgi()\" />");
    printf("</form></td></tr></table>");
}

```

```

void call_pUrvanipAwa_xvanxva (char *p1, char *s1, char *p2, char *s2) {
    int position;
    char sUwrastring[MEDIUMUTF];
    position = 1;
    sUwrastring[0] = '\0';
    if(rAjaxanwAxigaNa(p1,p2)) {
        /* exchange p1_utf,2 and also sup1,2 */

```

```

position=2;
strcpy(sUwrastring,"rAjaxanwAxiRu param 2.2.31");
/* printf("Case 1\n");*/
} else if(strcmp(p1,"ahan") && strcmp(p2,"rAwri")) {
    if(!GI(p1)){
        if(GI(p2)){
            position=2;
            strcpy(sUwrastring,"xvanxve Gi 2.2.32");
            /*printf("Case 2\n");*/
        } else if(strcmp(p1,"apara") && strcmp(p2,"para")){
            if(count_ac(p2)< count_ac(p1)){
                position=2;
                strcpy(sUwrastring,"alpAcwaram 2.2.34");
                /*printf("Case 4\n");*/
            } else {
                if(count_ac(p1)< count_ac(p2)){
                    position=1;
                    strcpy(sUwrastring,"alpAcwaram 2.2.34");
                    /*printf("Case 4\n");*/
                } else {
                    if(!ajAxi(p1) || !axanwa(p1)){
                        if(ajAxi(p2) && axanwa(p2)){
                            position=2;
                            strcpy(sUwrastring,"ajAxyaxanwam 2.2.33");
                            /*printf("Case 3\n");*/
                        }
                    }
                }
            }
        } else {strcpy(sUwrastring,"xvanxve Gi 2.2.32");}
    }
    printf("Within pUrvanipAwa_xvanxva position = %d",position);

```

```

    printf("<td><form name=\"f3\" method=\"post\" />");
    printf("<input type=\"hidden\" name=\"encodingpUrvanipAwa\"
id=\"encodingpUrvanipAwa\" value=\"%s\" />",encoding);
    printf("<input type=\"hidden\" name=\"vigrahapUrvanipAwa\"
id=\"vigrahapUrvanipAwa\" value=\"%s\" />",avigraha);
    printf("<input type=\"hidden\" name=\"p1pUrvanipAwa\"
id=\"p1pUrvanipAwa\" value=\"%s\" />",p1);
    printf("<input type=\"hidden\" name=\"p2pUrvanipAwa\"
id=\"p2pUrvanipAwa\" value=\"%s\" />",p2);

```

```

    printf("<input type=\"hidden\" name=\"s1pUrvanipAwa\"
id=\"s1pUrvanipAwa\" value=\"%s\" />",sup1);
    printf("<input type=\"hidden\" name=\"s2pUrvanipAwa\"
id=\"s2pUrvanipAwa\" value=\"%s\" />",sup2);
    printf("<input type=\"hidden\" name=\"samAsaparakArapUrvanipAwa\"
id=\"samAsaparakArapUrvanipAwa\" value=\"%s\" />",samAsaparakAra);
    printf("<input type=\"hidden\" name=\"samAsAnwapUrvanipAwa\"
id=\"samAsAnwapUrvanipAwa\" value=\"%s\" />",samAsAnwa);
    printf("<input type=\"hidden\" name=\"viXAyakasUwra\"
id=\"viXAyakasUwra\" value=\"%s\" />",viXAyakasUwra);
    printf("<input type=\"hidden\" name=\"positionpUrvanipAwa\"
id=\"positionpUrvanipAwa\" value=\"%d\" />",position);
    printf("<input type=\"hidden\" name=\"sUwrapUrvanipAwa\"
id=\"sUwrapUrvanipAwa\" value=\"%s\" />",sUwrastring);
    printf("<input type=\"hidden\" name=\"dividpUrvanipAwa\"
id=\"dividpUrvanipAwa\" value=\"#output3\" />");
    printf("<input type=\"submit\" name=\"anspUrvanipAwa\"
value=\"Continue\" id=\"anspUrvanipAwa\" onclick=\"return
pUrvanipAwaNullcgi()\"/>");
    printf("</form></td></tr></table>");
}

```

```

int rAjaxanwAxigaNa(char *p1,char *p2) {
    if(!strcmp(p1,"rAjan") && !strcmp(p2,"xanwa")) { return 1;}
    if(!strcmp(p2,"rAjan") && !strcmp(p1,"xanwa")) { return 1;}
    return 0;
}

```

```

int ajAxi(char *p){
    char ac[14],i;
    strcpy(ac,"aAiIuUqQLeEoO");
    if(index(ac,p[0])) { return 1;} else {return 0;}
}

```

```

int axanwa(char *p){
    if(p[strlen(p)-1] == 'a') { return 1;} else {return 0;}
}

```

```

int count_ac(char *p){
    int len,count;
    char ac[14],i;
    len = strlen(p);
    count=0;
}

```

```

strcpy(ac,"aAiLuUqQLeEoO");
for(i=0;i<len;i++){
    if(index(ac,p[i])) { count++;}
}
return count;
}

```

```

int GI(char *p) {
    char str[VERYSMALL];
    strcpy(str,"iLuU");
    if(index(str,p[strlen(p)-1])) { return 1;}
    else { return 0;}
}

```

Gati.lex

```
%{
#include "struct.h"
extern char ques[LARGE];
extern char p1_utf[MEDIUMUTF];
extern char p2_utf[MEDIUMUTF];
}%

prA [a-zA-Z]+

sup \+(su|O|jas|am|Ot|Sas|tA|ByAm3|Bis|fe|ByAm4|Byas4|fasi|ByAm5|Byas5|
fas|os6|Am|fi|os7|sup)
word {prA} {sup}

%%
^saw {sup} [ ] {word} {
    sprintf(ques,"किम् %s इति आदरवाची वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.62");
}
^alam [ ] {word} {
    sprintf(ques,"किम् %s इति भूषणार्थे वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.63");
}
^antar [ ] {word} {
    sprintf(ques,"किम् %s इति अपरिग्रहार्थे अस्ति वा (Y/N)?",
p1_utf);
    gawi_cgi_interface(ques,1,"1.4.64");
}
^kaNe [ ] {word} {
    sprintf(ques,"किम् %s इति श्रद्धायां वर्तते वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.65");
}
^manasi [ ] {word} {
    sprintf(ques,"किम् %s इति प्रतीघाते वर्तते वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.65");
}
^puras [ ] {word} {
    gawi_cgi_interface("NULL",1,"1.4.66");
}
^aswam [ ] {word} {
    sprintf(ques,"किम् %s इति आदरवाची वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.67");
}
```

```

    }
^acCa[ ](gawi|vada)    {
    sprintf(ques,"किम् %s इति गत्यर्थको वा (Y/N)?", p2_utf);
    gawi_cgi_interface(ques,1,"1.4.68");
}
^adas{sup}[ ]{word}    {
    sprintf(ques,"किम् %s इति अनुपदेशे वर्तते वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.69");
}
^wiras{sup}[ ]{word}    {
    sprintf(ques,"किम् %s इति अन्तर्धौ वर्तते वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.70");
}

^wiras{sup}[ ]kqwvA    {
    sprintf(ques,"किम् %s इति आदरवाची वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.72");
}
^(upAje|anvAje)[ ]kqwvA    {
    sprintf(ques,"किम् %s इति आदरवाची वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.73");
}
^(urasi|manasi)[ ]kqwvA {
    sprintf(ques,"किम् %s इति अनत्याधानार्थे वर्तते (Y/N)?",
p1_utf);
    gawi_cgi_interface(ques,1,"1.4.75");
}
^(maXye|paxe|nivecane)[ ]kqwvA    {
    gawi_cgi_interface("NULL",1,"1.4.76");
}
^(haswe|pANO)[ ]kqwvA    {
    sprintf(ques,"किम् %s इति औपम्ये वर्तते (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.77");
}
^prAXvam{sup}[ ]{word}    {
    sprintf(ques,"किम् %s इति बन्धने वर्तते वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.78");
}
^(jIvika|upaniRaw){sup}[ ]{word}    {
    sprintf(ques,"किम् %s इति औपम्ये वर्तते वा (Y/N)?", p1_utf);
    gawi_cgi_interface(ques,1,"1.4.79");
}

```

```

^(ku|pra|para|apa|sam|anu|ava|nis|nir|xus|xur|vi|A|ni|aXi|awi|api|su|ux|aBi|prawi|
para|upa)[ ]{word}      {
    gawi_cgi_interface("NULL",1,"2.2.18");
}
^{word}[ ](ku|pra|para|apa|sam|anu|ava|nis|nir|xus|xur|vi|A|ni|aXi|awi|api|su|ux|
aBi|prawi|para|upa)      {
    gawi_cgi_interface("NULL",2,"2.2.18");
}
^.*
{
    printf("No sUwra is applicable. Exiting ...\n");
    exit(0);
}

%%
extern char encoding[MEDIUM];
extern char avigraha[MEDIUM];
extern char p1[MEDIUM];
extern char p2[MEDIUM];
extern char p1_utf[MEDIUMUTF];
extern char p2_utf[MEDIUMUTF];
extern char sup1[MEDIUM];
extern char sup2[MEDIUM];
extern char samAsaprakAra[MEDIUM];
extern char viXAyakasUwra[LARGE];
extern char samAsAnwa[MEDIUM];

call_gawilex(char *in){
    yy_scan_string(in);
    gawilex();
}

gawiwrap()
{
}

gawi_cgi_interface(char *question,int position, char *sUwra){

    if (strcmp(question,"NULL"))
        printf("</tr></table><form name=\"f3\" method=\"post\">");
    else
        printf("<td><form name=\"f3\" method=\"post\">");

    printf("<input type=\"hidden\" name=\"encodinggawi\"
id=\"encodinggawi\" value=\"%s\" />",encoding);

```

```

        printf("<input type=\"hidden\" name=\"vigrahagawi\" id=\"vigrahagawi\"
value=\"%s\" />",avigraha);
        printf("<input type=\"hidden\" name=\"p1gawi\" id=\"p1gawi\"
value=\"%s\" />",p1);
        printf("<input type=\"hidden\" name=\"p2gawi\" id=\"p2gawi\"
value=\"%s\" />",p2);
        printf("<input type=\"hidden\" name=\"s1gawi\" id=\"s1gawi\"
value=\"%s\" />",sup1);
        printf("<input type=\"hidden\" name=\"s2gawi\" id=\"s2gawi\"
value=\"%s\" />",sup2);
        printf("<input type=\"hidden\" name=\"samAsaprakAragawi\"
id=\"samAsaprakAragawi\" value=\"%s\" />",samAsaprakAra);
        printf("<input type=\"hidden\" name=\"samAsAnwagawi\"
id=\"samAsAnwagawi\" value=\"%s\" />",samAsAnwa);
        printf("<input type=\"hidden\" name=\"viXAyakasUwragawi\"
id=\"viXAyakasUwragawi\" value=\"%s\" />",viXAyakasUwra);
        printf("<input type=\"hidden\" name=\"positiongawi\" id=\"positiongawi\"
value=\"%d\" />",position);
        printf("<input type=\"hidden\" name=\"gawisUwra\" id=\"gawisUwra\"
value=\"%s\" />",sUwra);
        if(strcmp(question,"NULL")){
            printf("%s",question);
            printf("<input type=\"submit\" name=\"ansgawi\" value=\"Yes\"
id=\"ansgawi\" onclick=\"return gawiyescgi()\" />");
            printf("<input type=\"submit\" name=\"ansgawi\" value=\"No\"
id=\"ansgawi\" onclick=\"return gawinocgi()\" />");
        } else if ((position == 1) || (position == 2)){
            if (!strcmp(encoding,"RMN")) {
                printf("<input type=\"submit\" name=\"ansgawi\"
value=\"Continue\" id=\"ansgawi\" onclick=\"return gawiNullcgi()\" />");
            } else {
                printf("<input type=\"submit\" name=\"ansgawi\"
value=\"Continue\" id=\"ansgawi\" onclick=\"return gawiNullcgi()\" />");
            }
        } else {
            printf("No sUwra is applicable. Exiting ...\n");
            exit(0);
        }
        if (strcmp(question,"NULL"))
            printf("</form>");
        else
            printf("</form></td></tr></table>");
    }
}

```

Aluk.lex

```
%{
#include "struct.h"
extern char ques[LARGE];
extern char alukviBARA[MEDIUM];
extern char sUwrastr[LARGE];
extern char p1_utf[MEDIUMUTF];
extern char p2_utf[MEDIUMUTF];
}%

%x aluk
HAL [hyvrlfmFNnJBgDXjbgdxKPCTWctwkpSRs]

prA [a-zA-Z]+
sup (su|O|jas|am|Ot|Sas|tA|ByAm3|Bis|fe|ByAm4|Byas4|fasi|ByAm5|Byas5|fas|
os6|Am|fi|os7|sup)

praWamA (su|O|jas)
wqwIyA (tA|ByAm3|Bis)
cawurWI (fe|ByAm4|Byas4)
paFcamI (fasi|ByAm5|Byas5)
RaRTI (fas|os6|Am)
sapwamI (fi|os7|sup)

word {prA}\+{sup}
%%

^(swoka|anwika|xUra|kqcCra|samIpa|aByASa)\+{paFcamI}
    {aluk_cgi_interface("NULL",1,"पञ्चम्याः स्तोकादिभ्यः(6.3.2)");}
^(ojas|sahas|amBas|wamas)\+{sup}[ ]{word}
    {aluk_cgi_interface("NULL",1,"ओजःसहोऽम्भस्तमसस्तृतीयायाः(6.3.3)");}
^manas\+{wqwIyA}[ ]AjFAyin\+{praWamA}
    {aluk_cgi_interface("NULL",1,"आज्ञायिनि च(6.3.5)");}
^manas\+{wqwIyA}[ ]{word}
    {
        sprintf(ques,"अनेन समासेन संज्ञा गम्यते
(Y/N)?");
        aluk_cgi_interface(ques,1,"मनसः
संज्ञायाम्(6.3.4)");
    }
^Awman\+{cawurWI}[ ](paxa|BARA)\+{praWamA}
    {aluk_cgi_interface("NULL",1,"वैयाकरणाख्यायाम् चतुर्थ्याः(6.3.7)");}
^Awman\+{sup}[ ]{word}
```

```

        {aluk_cgi_interface("NULL",1,"आत्मनश्च(6.3.6)");}
^para\+{cawurWI}[ ](paxa|BARA)\+{praWamA}
        {aluk_cgi_interface("NULL",1,"परस्य च(6.3.8)");}
^(.*a|. *{HAL})\+{sapwamI}[ ](kArRApaNa|mARaka|wripaxika)\+
{praWamA}    {
                                                                    sprintf(ques,"किं %s प्राक्देशकारार्थं
वर्तते एवं समासेन संज्ञा गम्यते(Y/N)?",p1_utf);

aluk_cgi_interface(ques,1,"कारनाम्नि च प्राचं हलादौ(6.3.10)");
}

^(prAvqR|Saraw|kAla|xiv)\+{sapwamI}[ ]ja\+{praWamA}
        {aluk_cgi_interface("NULL",1,"प्रावृट्छस्त्कालदिवां जे(6.3.15)");}
^(varRa|kRara|Sara|vara)\+{sapwamI}[ ]ja\+{praWamA}
{aluk_cgi_interface("NULL",2,"विभाषा वर्षक्षरशरवरात्(6.3.16)");}
^(.*a|. *{HAL})|araNya|vana)\+{sapwamI}[ ]{prA}\+{praWamA}    {
                                                                    sprintf(ques,"अनेन समासेन
संज्ञा गम्यते(Y/N)?");

aluk_cgi_interface(ques,1,"हलदन्तात्सप्तम्याः संज्ञायाम्(6.3.9)");
}

^maXya\+{sapwamI}[ ]guru\+{praWamA}
        {aluk_cgi_interface("NULL",1,"मध्याद् गुरौ(6.3.11)");}
^(.*a|. *{HAL})\+{sapwamI}[ ]banXa\+{praWamA}
        {aluk_cgi_interface("NULL",1,"बन्धे च विभाषा(6.3.13)");}
/* 6.3.14 wawpuruRe kqwi bahulam */
/* 6.3.16
^[ ](warap|wamap|kAla|wana) {strcpy(tmp,"0"); strcpy(sUwrastr,"घकालतनेषु
कालनाम्नः(6.3.17)"); BEGIN 0;}
/* Check how to write not equal to */
^{prA}\+{sapwamI}[ ](wara|wama|kAla|wana)\+{sapwamI}
{aluk_cgi_interface("NULL",2,"घकालतनेषु कालनाम्नः(6.3.17)");}
^{prA}\+{sapwamI}[ ](Saya|vAsa|vAsI)\+{praWamA}    {
                                                                    sprintf(ques,"किमिदं %s
कालभिन्नवाचकः(Y/N)?",p1_utf);

aluk_cgi_interface(ques,2,"शयवासवासिष्वकालात्(6.3.18)");
}

^{prA}\+{sapwamI}[ ](. *in|sixXa|baxXa)\+{praWamA}
        {aluk_cgi_interface("NULL",0,"नेन्सिद्धबध्नातिषु च(6.3.19)");}
^{prA}\+{sapwamI}[ ]{prA}\+{praWamA}    {
                                                                    sprintf(ques,"किमिदं %s स्वाङ्गवाचकं वर्तते एवं
समासेन संज्ञा प्रतीयते(Y/N)?",p1_utf);

```

```

aluk_cgi_interface(ques,1,"अमूर्धमस्तकात्
स्वाङ्गात् अकामे(6.3.12)");
    }
^{\prA}\+{\sapwamI}[ ]{\word}
    {
    sprintf(ques,"किमिदं %s भाषायां वर्तते
(Y/N)?",p1_utf);
    aluk_cgi_interface(ques,0,"स्थे च
भाषायाम्(6.3.20)");
    }

^{\prA}\+{\RaRTI}[ ]puwra\+{\praWamA}
    {
    sprintf(ques,"किं समासेन निन्दार्थः
गम्यते(Y/N)?");
    aluk_cgi_interface(ques,2,"पुत्रेऽन्यतरस्याम्(6.3.22)");
    }

^{\prA}q\+{\RaRTI}[ ]puwra\+{\praWamA}
    {
    sprintf(ques,"किं %s विद्यासम्बन्धार्थे
वर्तते अथवा योनिसम्बन्धार्थे(Y/N)?",p1_utf);
    aluk_cgi_interface(ques,1,"ऋतो
विद्यायोनिसम्बन्धेभ्यः(6.3.23)");
    }

^{\prA}q\+{\RaRTI}[ ](svasq|pawi)\+{\praWamA}
    {
    aluk_cgi_interface("NULL",2,"विभाषा स्वसृपत्योः(6.3.24)");
    }

^{\prA}\+{\RaRTI}[ ]{\word}
    {
    sprintf(ques,"किम् अनेन समासेन आक्रोशार्थः
गम्यते(Y/N)?");
    aluk_cgi_interface(ques,1,"षष्ठ्या
आक्रोशे(6.3.21)");
    }

^.*
    {aluk_cgi_interface("NULL",0,"NULL");}

%%

char ques[LARGE];
char p1_utf[MEDIUMUTF];
char p2_utf[MEDIUMUTF];
char p1_utfr[MEDIUMUTF];
char p2_utfr[MEDIUMUTF];

```

```

char p1[MEDIUM];
char p2[MEDIUM];
char sup1[MEDIUM];
char sup2[MEDIUM];
char samAsaprakAra[MEDIUM];
char viXayakasUwra[LARGE];
char samAsAnwa[MEDIUM];
char avigraha[MEDIUM];
char encoding[MEDIUM];
int divid;

main (int argc, char *argv[]) {
    strcpy(encoding,argv[1]);
    strcpy(avigraha,argv[2]);
    strcpy(p1,argv[3]);
    strcpy(sup1,argv[4]);
    strcpy(p2,argv[5]);
    strcpy(sup2,argv[6]);
    strcpy(samAsaprakAra,argv[7]);
    strcpy(samAsAnwa,argv[8]);
    divid = atoi(argv[9]);

    cnvrtwx2utf(p1,p1_utf);
    cnvrtwx2utf(p2,p2_utf);

    if(!strcmp(encoding,"RMN")){
        cnvrtwx2utf(p1_utf,p1_utfr);
        cnvrtwx2utf(p2_utf,p2_utfr);

        strcpy(p1_utf, p1_utfr);
        strcpy(p2_utf, p2_utfr);
    }
    // printf("aluk.lex samAsAnwa = %s",samAsAnwa);
    call_aluklex(avigraha);
}

call_aluklex(char *in){
    yy_scan_string(in);
    yylex();
}

yywrap()
{

```

```

}
aluk_cgi_interface (char *question, int alukviBARA, char *sUwra)
{
    char questionr[MEDIUMUTF];
    // printf("samAsAnwa within aluk_cgi_interface = %s",samAsAnwa);
    if(strcmp(question,"NULL"))
        printf("</tr></table><form name=\"f4\" method=\"post\">");
    else
        printf("<td><form name=\"f4\" method=\"post\">");
    printf("<input type=\"hidden\" name=\"encodingaluk\" id=\"encodingaluk\"
value=\"%s\" />",encoding);
    printf("<input type=\"hidden\" name=\"p1aluk\" id=\"p1aluk\" value=\"%s\"
/>",p1);
    printf("<input type=\"hidden\" name=\"p2aluk\" id=\"p2aluk\" value=\"%s\"
/>",p2);
    printf("<input type=\"hidden\" name=\"s1aluk\" id=\"s1aluk\"
value=\"%s\" />",sup1);
    printf("<input type=\"hidden\" name=\"s2aluk\" id=\"s2aluk\"
value=\"%s\" />",sup2);
    printf("<input type=\"hidden\" name=\"samAsaprakAraaluk\"
id=\"samAsaprakAraaluk\" value=\"%s\" />",samAsaprakAra);
    printf("<input type=\"hidden\" name=\"samAsAnwaaluk\"
id=\"samAsAnwaaluk\" value=\"%s\" />",samAsAnwa);
    printf("<input type=\"hidden\" name=\"dividaluk\" id=\"dividaluk\"
value=\"#output%d\" />",divid);
    printf("<input type=\"hidden\" name=\"alukviBARA\" id=\"alukviBARA\"
value=\"%d\" />",alukviBARA);
    printf("<input type=\"hidden\" name=\"sUwraaluk\" id=\"sUwraaluk\"
value=\"%s\" />",sUwra);
    if(strcmp(question,"NULL")){
        if(!strcmp(encoding,"RMN")){
            cnvrtutfd2r(question,questionr);
            printf("%s ",questionr);
        } else {
            printf("%s ",question);
        }
    }
    printf("<input type=\"submit\" name=\"ansaluk\" value=\"Yes\"
id=\"ansaluk\" onclick=\"return alukyescgi()\"/>");
    printf("<input type=\"submit\" name=\"ansaluk\" value=\"No\"
id=\"ansaluk\" onclick=\"return aluknocgi()\"/>");
    } else
        if (!strcmp(encoding,"RMN")) {
            printf("<input type=\"submit\" name=\"ansaluk\" value=\"Continue\"

```

```

id=\"ansaluk\" onclick=\"return alukNullcgi()\"/>");
    } else {
        printf("<input type=\"submit\" name=\"ansaluk\" value=\"Continue\"
id=\"ansaluk\" onclick=\"return alukNullcgi()\"/>");
    }
    if(strcmp(question,"NULL"))
        printf("</form>");
    else
        printf("</form></td></tr></table>");
}

```

Purvapada.lex

```
%{
#include "struct.h"

extern char ques[LARGE];
extern char paxa[LARGE];
extern char sUwrastr[LARGE];
extern char p1_utf[MEDIUMUTF];
extern char p2_utf[MEDIUMUTF];
extern struct prAwipaxikam prAwi;
extern char p1[MEDIUM];
extern char p2[MEDIUM];
%}

ac [aAiLuUqQleEoO]
HAL [hyvrlfmFNnJBgDXjbgdxKPCTWctwkpSRs]
prA [a-zA-Z]+
sup (su|O|jas|am|Ot|Sas|tA|ByAm3|Bis|fe|ByAm4|Byas4|fasi|ByAm5|Byas5|fas|
os6|Am|fi|os7|sup)

saMKyAbahuvrIhyaSIwi(ekam|xve|wrINi|wri|cawvAri|paFca|RaRTa|sapwa|
aRTa|nava|xaSa|ekAxaSa|xvAxaSa|wrayoxaSa|cawurxaSa|paFcaxaSa|RodaSa|
sapwaxaSa|aRTAxaSa|navaxaSa|viMSawi|ekaviMSawi|xvAviMSawi|
wrayoviMSawi|cawurviMSaWi|paFcaviMSawi|RadviMSaWi|sapwaviMSawi|
aRtAviMSawi|navaviMSawi|wriMSaw|ekawriMSaw|xvAwriMSaw|
wrayaswriMSaw|cawuswriMSaw|paFcawriMSaw|RatwriMSaw|
sapwawriMSaw|aRTAwriMSaw|navawriMSaw|cawvAriMSaw|
ekacawvAriMSaw|xvicawvAriMSaw|wricawvAriMSaw|cawuscawvAriMSaw|
paFcacawvAriMSaw|RatcawvAriMSaw|sapwacawvAriMSaw|
aRtAcawvAriMSaw|navacawvAriMSaw|paFcASaW)

%x TMP

%%
^{prA}[vy]/[vrl].* {strncpy(paxa,yytext,yytext-1); strcpy(sUwrastr,"लोपो
व्योर्वलि 6.1.66"); BEGIN TMP;}
<TMP>{prA}[ ]{prA} {strcat(paxa,yytext);
pUrvapaxa_cgi_interface("NULL",paxa,sUwrastr);}
^(mahawI|mahaw)[ ]{prA}.*
    {pUrvapaxa_cgi_interface("NULL","mahA","", "आन्महतः
समानाधिकरणजातीययोः 6.3.46"); }
/* Though the sUwra has the word mahaw, it is to be interpreted as
```

mahaw/mahawI, and w -> A, followed by a sandhi rule a+A -> A. All these steps are compressed into one. */

```
^uRas[ ]{prA}[ ]xvanxva {
    sprintf(ques,"अनेन समासेन देवतार्थः उच्यते(Y/N)?");
    pUrvapaxa_cgi_interface(ques,"uRAsA",p1_utf,"उषासोषसः
6.3.31");
}
^agni[ ](soma|varuNa).*
    {pUrvapaxa_cgi_interface("NULL","agnI",p1_utf,"ईदग्रेः सोमवरुणयोः
6.3.27");}
^mAwaq[ ]piwq
    {pUrvapaxa_cgi_interface("NULL","mAwa",p1_utf,"मातरपितरवुदीचाम्
6.3.32");}
^xvi[ ]aSIwi[ ]bahuvrIhi
    {pUrvapaxa_cgi_interface("NULL","xvA",p1_utf,"द्व्यष्टनः
संख्यायामबहुव्रीह्यशीत्योः 6.3.47");}
^xiv[ ]pqWivI.*
    {pUrvapaxa_cgi_interface("NULL","xivAs/xyAvA",p1_utf,"दिवसश्च
पृथिव्याम् 6.3.30");}
^xiv[ ]{prA}.*
    {pUrvapaxa_cgi_interface("NULL","xyAvA",p1_utf,"दिवो द्यावा
6.3.29");}
^xvi[ ]{saMKyAbahuvrIhyaSIwi}.*
    {pUrvapaxa_cgi_interface("NULL","xvA",p1_utf,"द्व्यष्टनः
संख्यायामबहुव्रीह्यशीत्योः 6.3.47");}
^aRtan[ ]{saMKyAbahuvrIhyaSIwi}.*
    {pUrvapaxa_cgi_interface("NULL","aRtA",p1_utf,"द्व्यष्टनः
संख्यायामबहुव्रीह्यशीत्योः 6.3.47");}
^wri[ ]{saMKyAbahuvrIhyaSIwi}.*
    {pUrvapaxa_cgi_interface("NULL","wrayas",p1_utf,"त्रेस्त्रयः 6.3.48");}
^hqxaya[ ](leKa|lAsa).*
    {pUrvapaxa_cgi_interface("NULL","hqw",p1_utf,"हृदयस्य हृल्लेखयदण्लासेषु
6.3.50");} /* to add yax and aN */
^hqxaya[ ](Soka|roga).*
    {pUrvapaxa_cgi_interface("NULL","hqw",p1_utf,"वा शोकष्यत्रोगेषु
6.3.51");}
^pAxa[ ](ga|Aji|Awi|upahawa).*
    {pUrvapaxa_cgi_interface("NULL","paxa",p1_utf,"पादस्य
पदाज्यातिगोपहतेषु 6.3.52");}
/* paxyawyawaxarWe 6.3.53 waxXiwa Morph is needed */
^pAxa[ ](hima|kARi|hawi).*
    {pUrvapaxa_cgi_interface("NULL","pax",p1_utf,"हिमकाषिहतिषु च
6.3.54");}
```

```

^pAxa[ ](GoRa|miSra|Sabxa).*
    {pUrvapaxa_cgi_interface("NULL","paw",p1_utf,"वा घोषमिश्रशब्देषु
6.3.56");}
^uxaka[ ](manWa|oxana|sakwu|binxu|vajra|BAra|vIvaXa|gAha).*

    {pUrvapaxa_cgi_interface("NULL","uxa",p1_utf,"मन्थौदनसक्तुबिन्दुवज्रभारवीवधगा
हेषु च 6.3.60");}
^uxaka[ ]{prA}[ ]wawpuruRa {
    sprintf(ques,"अनेन समासेन सञ्ज्ञा गम्यते(Y/N)?");

pUrvapaxa_cgi_interface(ques,"uxa",p1_utf,"उदकस्योदः सञ्ज्ञायाम् 6.3.57");
    }
^uxaka[ ](peRa|vAsa|vAhana|Xi).*
    {pUrvapaxa_cgi_interface("NULL","uxa",p1_utf,"पेषंवासवाहनधिषु च
6.3.58");}
/* ekahalAxO pUrayiwavyeZnyawarasyAm 6.3.59 how to implement
ekahalAxO?*/
/* iko hrasvoZfyo gAlavasya 6.3.61 morph */
^ekA[ ]{prA}.*      {pUrvapaxa_cgi_interface("NULL","eka",p1_utf,"एक
तद्धिते च 6.3.62");}
/* fyApoH saFjFACanxasorbahulam 6.3.63 morph information */
/* wve ca 6.3.64 waxXiwa suffix    Not Implemented */
^iRtakA[ ]ciwa.*
    {pUrvapaxa_cgi_interface("NULL","iRtaka",p1_utf,"इष्टकेषिकामालानाम्
चित्तूलभारिषु 6.3.65");} /* yatha samkhya */
^iRIkA[ ]wUla.*
    {pUrvapaxa_cgi_interface("NULL","iRIka",p1_utf,"इष्टकेषिकामालानाम्
चित्तूलभारिषु 6.3.65");}
^mAIA[ ]BArI.*
    {pUrvapaxa_cgi_interface("NULL","mAIA",p1_utf,"इष्टकेषिकामालानाम्
चित्तूलभारिषु 6.3.65");}
/* 6.3.66 to 6.3.69 are not related to samAsAs */
^sawya[ ]kAra.*
    {pUrvapaxa_cgi_interface("NULL","sawyam",p1_utf,"कारे सत्यागदस्य
6.3.70");} /* sUwrastr is split */
^agaxa[ ]kAra.*
    {pUrvapaxa_cgi_interface("NULL","agaxam",p1_utf,"कारे सत्यागदस्य
6.3.70");}
^rAwri[ ]cara.*      {
    if(chk_kqw(p2,"kwa")){
        pUrvapaxa_cgi_interface("NULL","rAwriM",p1_utf,"रात्रेः
कृति विभाषा 6.3.72");
    }
}

```

```

    }

/* 6-3-75
naBrANnapAnnaveXAnAsawyAnamucinakulanaKanapuMsakanakRawranakra
nAkeRu prakqwyA */
/* 6-3-76 ekAxiH ca Ekasya ca xuk */
^na[ ]ga[ ]wawpuruRa {
    sprintf(ques,"अनेन समासेन प्राणिभिन्नः उच्यते(Y/N)?");
    pUrvapaxa_cgi_interface(ques,"na","a","नगः अप्राणिषु
अन्यतरस्याम् 6.3.77");
}
^na[ ]{HAL}.* {pUrvapaxa_cgi_interface("NULL","a",p1_utf,"नलोपो नञः
6.3.73");}
^na[ ]{ac}.* {pUrvapaxa_cgi_interface("NULL","an",p1_utf,"तस्मान्नुडचि
6.3.74");}

^saha[ ]{prA}[ ]avyayIBAvAH {
    sprintf(ques,"अनेन समासेन सञ्ज्ञा गम्यते(Y/N)?");
    pUrvapaxa_cgi_interface(ques,"sa",p1_utf,"सहस्य सः
सञ्ज्ञायाम् 6.3.78",paxa,sUwrastr,20);
    sprintf(ques,"%s इति पदं कालभिन्नवाचकम्(Y/N)?",p2_utf);
    pUrvapaxa_cgi_interface("NULL","sa",p1_utf,"अव्ययीभावे
च अकाले 6.3.81");
    pUrvapaxa_cgi_interface("NULL","saha",p1_utf,"अव्ययीभावे च अकाले
6.3.81"); /* else */
}

/* 6-3-79 granWAnwAXike ca */
/* 6-3-80 xviwIye ca anupAKye */
/* 6-3-82 vA upasarjanasya */
/* 6-3-83 prakqwyA ASiRi */
/* 6-3-84 samAnasya Canxasi amUrXapraBqwyuxarkeRu */
^samAna[ ](jyowis|janapaxa|rAwri|nABi|nAma|gowra|rUpa|sWAna|varNa|vaya|
vacana|banXu).*

{pUrvapaxa_cgi_interface("NULL","sa",p1_utf,"ज्योतिर्जनपदरात्रिनाभिनामगोत्ररूप
स्थनवर्णवयोवचनबन्धुषू 6.3.85");}
^samAna[ ]brahmacArI.*
    {pUrvapaxa_cgi_interface("NULL","sa",p1_utf,"चरणे ब्रह्मचारिनि
6.3.86");}
/* 6-3-87 wIrWe ye morph*/
/* 6-3-88 viBARA uxare morph*/

```

```

^samAna[ ](xqk|xqS).*
    {pUrvapaxa_cgi_interface("NULL","sa",p1_utf,"दृग्दृशवतुषु 6.3.89");}
^ixam[ ](xqk|xqS).*
    {pUrvapaxa_cgi_interface("NULL","I",p1_utf,"इदङ्गिमोरीशकी 6.3.90");}
^kim[ ](xqk|xqS).*
    {pUrvapaxa_cgi_interface("NULL","kI",p1_utf,"इदङ्गिमोरीशकी 6.3.90");}
/* 6-3-97 xvyanwarupasargeByaH apaH Iw */
/* 6-3-98 UxanoH xeSe */
^anya[ ](ASis|ASA|AswA|AsWiwa|uwsuka|Uwi|kAraka|rAga).*

{pUrvapaxa_cgi_interface("NULL","anyax",p1_utf,"अषष्ठ्यतृतीयास्थान्यस्य
दुगाशीराशास्त्रास्थितोत्सुकोतिकारकरागच्छेषु 6.3.99");}
/* 6-3-100 arWe viBARA Not Implemented */
/* 6-3-105 IRaxarWe */
/* 6-3-109 pqRoxarAxIni yaWopaxiRtam how to implement this one?*/
/* 6-3-114 saMhiwAyAm */
/* 6-3-115 karNe lakRaNasya
aviRtARtapaFcamaNiBinnacCinnacCixrasruvasvaswikasya negation */
/* 6-3-116 nahivqwivqRivyaXirucisahiوانيRu kvO morph */
/* 6-3-117 vanagiryOH saFjFAyAM kotarakiMSulukAxInAm */
/* 6-3-118 vale morph*/
/* 6-3-119 mawO bahvacaH anajirAxInAm morph*/
/* 6-3-120 SarAxInAM ca morph*/
/* 6-3-121 ikaH vaheH apIloH morph */
/* 6-3-122 upasargasya GaFi amanuRye bahulam morph*/
/* 6-3-125 aRtanaH saFjFAyAm */
/* 6-3-129 nare saFjFAyAm */
/* 6-3-139 samprasAraNasya */
^anya[ ]arWa.*
    {pUrvapaxa_cgi_interface("NULL","anyax",p1_utf,"अर्थे विभाषा
6.3.100");}
^ku[ ](raWa|vaxa).*
    {pUrvapaxa_cgi_interface("NULL","kax",p1_utf,"रथवदयोश्च 6.3.102");}
^ku[ ]wqNa.*
    {pUrvapaxa_cgi_interface("NULL","kax",p1_utf,"तृणे
च जातो 6.3.103");}
^ku[ ](paWi|akRa).*
    {pUrvapaxa_cgi_interface("NULL","kA",p1_utf,"कापथ्यक्षयोः 6.3.104");}
^ku[ ]puruRa.*
    {pUrvapaxa_cgi_interface("NULL","kA",p1_utf,"विभाषा पुरुषै 6.3.106");}
^ku[ ](uRNa).*
    {pUrvapaxa_cgi_interface("NULL","kava/kA/kax",p1_utf,"कवं चोष्णे
6.3.107");}
^ku[ ]{ac}.* {pUrvapaxa_cgi_interface("NULL","kax",p1_utf,"कोः कत्तत्पुरुषेऽचि

```

```

6.3.101");}
^(vi|ni|aXi|awi|api|aBi|prawi)[ ]kASa.*      {

pUrvapaxa_cgi_interface("NULL","vI/NI/aXI/apI/aBI/prawI/awI",p1_utf,"इकः
काशे 6.3.123");
    }
^viSva[ ]miwra[ ]bahuvrIhi      {
    sprintf(ques,"अनेन समासेन ऋषिः उच्यते(Y/N)?");
    pUrvapaxa_cgi_interface(ques,"viSvA",p1_utf,"मित्रे
चर्षो 6.3.130");
    }
^ku[ ]wumburu.*
    {pUrvapaxa_cgi_interface("NULL","kus",p1_utf,"कुस्तुम्बुरुणि जातिः
6.1.143");}
^apara[ ]para.*
    {pUrvapaxa_cgi_interface("NULL","aparas",p1_utf,"अपरस्पराः
क्रियासातत्ये 6.1.144");}
^go[ ]paxa.* {pUrvapaxa_cgi_interface("NULL","goR",p1_utf,"गोष्पदम्
सेवितासेवितप्रमाणेषु 6.1.145");}
^A[ ]paxa.* {pUrvapaxa_cgi_interface("NULL","As",p1_utf,"आस्पदम्
प्रतिष्ठायाम् 6.1.146");}
^A[ ]carya.* {pUrvapaxa_cgi_interface("NULL","As",p1_utf," आश्चर्यमनित्ये
6.1.147");}
^ava[ ]kara.*
    {pUrvapaxa_cgi_interface("NULL","avas",p1_utf,"वर्चस्केऽवस्करः
6.1.148");}
^apa[ ]kara.*      {pUrvapaxa_cgi_interface("NULL","apas",p1_utf,"अपस्करो
स्थानाम् 6.1.149");}
^vi[ ]kira.* {pUrvapaxa_cgi_interface("NULL","viR",p1_utf,"विष्किरः शकुनौ वा
6.1.150");}
^prawi[ ]kaSa.*
    {pUrvapaxa_cgi_interface("NULL","prawiR",p1_utf,"प्रतिष्कशश्च कशेः
6.1.152");}
^pra[ ]kaNva.*
    {pUrvapaxa_cgi_interface("NULL","pras",p1_utf,"प्रस्कण्वहरिश्चन्द्रवृषी
6.1.153");} /* Note: sUwrastr is split */
^hari[ ]canxra.*
    {pUrvapaxa_cgi_interface("NULL","haris",p1_utf,"प्रस्कण्वहरिश्चन्द्रवृषी
6.1.153");}

^ku[ ]{prA}[ ]wawpuruRa {
    sprintf(ques,"किम् कु इति ईषदर्थे वर्तते(Y/N)?");
    pUrvapaxa_cgi_interface(ques,"kA",p1_utf,"ईषदर्थे 6.3.105");
}

```

```

    }
^kA[ ]wIra[ ]bahuvrIhi {
    sprintf(ques,"अनेन समासेन नगरार्थः प्रतीयते(Y/N)?");
    /* Is it a saFjFA or any nagara ? */
    pUrvapaxa_cgi_interface(ques,"kAs",p1_utf,"कास्तीराजस्तुन्दे
नगरे 6.1.155");
}
^aja[ ]wunxa[ ]wawpuruRa {
    sprintf(ques,"अनेन समासेन नगरार्थः प्रतीयते(Y/N)?");

pUrvapaxa_cgi_interface(ques,"ajas",p1_utf,"कास्तीराजस्तुन्दे नगरे 6.1.155");
}
^kAra[ ]kara.*
    {pUrvapaxa_cgi_interface("NULL","kAras",p1_utf,"कारस्करो वृक्षः
6.1.156");}
^(pAra|kAra)[ ]kara.* {strcpy(p1_utf,paxa); strcat(paxa,"s");
pUrvapaxa_cgi_interface("NULL",paxa,"","पारस्करप्रभृतीनि संज्ञायाम् 6.1.157");}
^{\prA}q[ ].* {
    strcpy(paxa,p1_utf);
    paxa[strlen(paxa)-1] = 'A';
    strcpy(sUwrastr,"आनङ्गतो द्वन्द्वे 6.3.25");
    pUrvapaxa_cgi_interface("NULL",paxa,p1_utf,sUwrastr);
}
^{\prA}[ ].* {
    strcpy(paxa,p1_utf);
    prAwipaxika_info(paxa,&prAwi);
    if(!strcmp(prAwi.jAwi,"xevawA")) {
        paxa[strlen(paxa)-1] = 'A';
        strcpy(sUwrastr,"देवता द्वन्द्वे च 6.3.26");

pUrvapaxa_cgi_interface("NULL",paxa,p1_utf,sUwrastr);
    } else
        pUrvapaxa_cgi_interface("NULL","", "");
    }

%%
extern int divid;
extern char samAsAnwa[MEDIUM];
extern char samAsaprakAra[MEDIUM];
extern char encoding[MEDIUM];
extern char p1[MEDIUM];
extern char p2[MEDIUM];
extern char p1_utf[MEDIUMUTF];

```

```

extern char p2_utf[MEDIUMUTF];

char paxa[LARGE];
char sUwrastr[LARGE];
char ques[LARGE];

#define myPATH "/home/pavan/scl/"

call_pUrvapaxalex(){
    char in[MEDIUM];
    strcpy(in,p1);
    strcat(in," ");
    strcat(in,p2);
    strcat(in," ");
    strcat(in,samAsaprakAra);
    yy_scan_string(in);
    pUrvapaxalex();
    //pUrvapaxa_cgi_interface("NULL","", "");
}

pUrvapaxawrap()
{
}

pUrvapaxa_cgi_interface(char *condition, char *pstrmod, char *pstrunmod,
char *sstr){

    char conditionr[LARGE];
    if(strcmp(condition,"NULL"))
        printf("</tr></table><form name=\"f5\" method=\"post\" />");
    else
        printf("<td><form name=\"f5\" method=\"post\" />");
        printf("<input type=\"hidden\" name=\"encodingpUrva\"
id=\"encodingpUrva\" value=\"%s\" />",encoding);
        printf("<input type=\"hidden\" name=\"p1pUrva\" id=\"p1pUrva\"
value=\"%s\" />",p1);
        printf("<input type=\"hidden\" name=\"p2pUrva\" id=\"p2pUrva\"
value=\"%s\" />",p2);
        printf("<input type=\"hidden\" name=\"samAsaprakArapUrva\"
id=\"samAsaprakArapUrva\" value=\"%s\" />",samAsaprakAra);
        printf("<input type=\"hidden\" name=\"samAsAnwapUrva\"
id=\"samAsAnwapUrva\" value=\"%s\" />",samAsAnwa);
        printf("<input type=\"hidden\" name=\"dividpUrva\" id=\"dividpUrva\"

```

```

value="#output%d\" />",divid);
    printf("<input type=\"hidden\" name=\"strmodpUrva\" id=\"strmodpUrva\"
value=\"%s\" />",pstrmod);
    printf("<input type=\"hidden\" name=\"strunmodpUrva\"
id=\"strunmodpUrva\" value=\"%s\" />",pstrunmod);
    printf("<input type=\"hidden\" name=\"sUwrastrpUrva\"
id=\"sUwrastrpUrva\" value=\"%s\" />",sstr);
    if(strcmp(condition,"NULL")){
        if(!strcmp(condition,"RMN")){
            cnvrtutfd2r(condition,conditionr);
            printf("%s",conditionr);
        } else {
            printf("%s",condition);
        }
        printf("<input type=\"submit\" name=\"anspUrva\" value=\"Yes\"
id=\"anspUrva\" onclick=\"return pUrvapaxayescgi()\" />");
        printf("<input type=\"submit\" name=\"anspUrva\" value=\"No\"
id=\"anspUrva\" onclick=\"return pUrvapaxanocgi()\" />");
    }
    else
        printf("<input type=\"submit\" name=\"anspUrva\" value=\"Continue\"
id=\"anspUrva\" onclick=\"return pUrvapaxaNullcgi()\" />");
    if(strcmp(condition,"NULL"))
        printf("</form>");
    else
        printf("</form></td></tr></table>");
}

int chk_kqw(char *str, char *kqw_str){
FILE *fp;
char ans[VERYSMALL];
char cmd[MEDIUM];
char fin[MEDIUM];
char fout[MEDIUM];
int pid;

pid = getpid();

sprintf(fin,"/tmp/tmp%d",pid);
sprintf(fout,"/tmp/result%d",pid);

fp = fopen(fin,"w");
fprintf(fp,"%s\n",str);

```

```

fclose(fp);

sprintf(cmd, "/usr/bin/lt-proc -c %s/morph_bin/skt_morf.bin < %s | grep -c
'kqw_prawyayaH:%s' > %s", myPATH, fin, kqw_str, fout);
system(cmd);

fp = fopen(fout, "r");
fscanf(fp, "%c", ans);
fclose(fp);

sprintf(cmd, "rm %s %s", fin, fout);
system(cmd);
if(ans[0] == '1') { return 1; } else { return 0;}
}

```

Uttarapada.lex

```
%{
#include "struct.h"
extern char ques[LARGE];
extern char ustr[MEDIUMUTF];
extern char sUwrastr[LARGE];
}%

prA [a-zA-Z]+
safKyA      (ekam|xve|wrINi|cawvAri|paFca|RaRTa|sapwa|aRTa|nava|xaSa|
ekAxaSa|xvAxaSa|wrayoxaSa|cawurxaSa|paFcaxaSa|RodaSa|sapwaxaSa|
aRTAxaSa|navaxaSa|viMSawi|ekaviMSawi|xvAviMSawi|wrayoviMSawi|
cawurviMSaWi|paFcaviMSawi|RadviMSaWi|sapwaviMSawi|aRtAviMSawi|
navaviMSawi|wriMSaw|ekawriMSaw|xvAwriMSaw|wrayaswriMSaw|
cawuswriMSaw|paFcawriMSaw|RatwriMSaw|sapwawriMSaw|aRTAwriMSaw|
navawriMSaw|cawvAriMSaw|ekacawvAriMSaw|xvicawvAriMSaw|
wricawvAriMSaw|cawuscawvAriMSaw|paFcacawvAriMSaw|
RateawvAriMSaw|sapwacawvAriMSaw|aRtAcawvAriMSaw|
navacawvAriMSaw|paFcASaW)
haswyAxi (haswin|kuxxAla|aSva|kaSika|kuruwa|katola|katolaka|gaNdola|
gaNdolaka|kaNdola|kaNdolaka|aja|kapowa|jAla|gaNda|mahilA|xAsI|gaNikA|
kusUla)
kumBAxi  (kumBa|eka|jAla|SUla|muni|guNa|Sawa|sUwra|goXA|kalaSI|vi|
wqNa|xvi|wri|Rat|xAsI|Siwi|viRnu|su|niR|Arxra|kuNi|kqRNa|Suci|xroNI|xroNa|
xru|sUkara|Sakqw|aRTA|sWUNA|a|sUcI)
%%

^akRi[ ]BrU[ ].* {uwwarapaxa_cgi_interface("NULL","Bruva","अचतुरविचतुर..
5.4.77");}
^({safKyA}|su)[ ]xanwa[ ]bahuvrIhi {
    sprintf(ques,"अनेन समासेन वयः उच्यते(Y/N)?");
    uwwarapaxa_cgi_interface(ques,"xawq","वयसि दन्तस्य
दत् 5.4.141");
}
^(agra|SuxXa|SuBra|vqRa|varAha)[ ]xanwa[ ].*
    {uwwarapaxa_cgi_interface("NULL","xawq","अग्रान्तशुद्धशुभ्रवृषवराहेभ्यश्च
5.4.145");}

^(SyAva|aroka)[ ]xanwa[ ].*
    {uwwarapaxa_cgi_interface("NULL","xawq","विभाषा श्यावरोकाभ्याम्
5.4.144");}
```

```

^{\prA}[ ]xanwa[ ]bahuvrIhi {
    sprintf(ques,"अनेन समासेन सञ्ज्ञा एवं स्त्रीत्वं गम्यते(Y/N)?");
    uwwarapaxa_cgi_interface(ques,"xawq","स्त्रियाम् सञ्ज्ञायाम्
5.4.143");
}
^wri[ ]kakuxa[ ]bahuvrIhi {
    sprintf(ques,"किमनेन समासेन पर्वतः उच्यते(Y/N)?");
    uwwarapaxa_cgi_interface(ques,"kakux","त्रिककुत्पर्वते
5.4.147");
}
^{\prA}[ ]kakuxa[ ]bahuvrIhi {
    sprintf(ques,"किम् अनेन समासेन अवस्था उच्यते(Y/N)?");
    uwwarapaxa_cgi_interface(ques,"kakux","ककुदस्यावस्थायाम् लोपः 5.4.146");
}
^(uw|vi)[ ]kAkuxa[ ].*
    {uwwarapaxa_cgi_interface("NULL","kAkux","उद्विभ्याम्काकुदस्य
5.4.148");}

^pUrNa[ ]kAkuxa[ ].*
    {uwwarapaxa_cgi_interface("NULL","kAkux","पूर्णाद्विभाषा 5.4.149");}

^sWulA[ ]nAsikA[ ].*
    {uwwarapaxa_cgi_interface("NULL","", "अञ्जासिकायाः संज्ञायाम् नसं
चास्थुलात् 5.4.118");}
^(xru|Kara)[ ]nAsikA[ ].*
    {uwwarapaxa_cgi_interface("NULL","nasa","अञ्जासिकायाः संज्ञायाम् नसं
चास्थुलात् 5.4.118");}
^(pra|para|apa|sam|anu|ava|nis|nir|xus|xur|vi|Af|ni|aXi|awi|api|su|ux|aBi|prawi|
pari|upa)[ ]nAsikA[ ].* {uwwarapaxa_cgi_interface("NULL","nasa","उपसर्गाच्च
5.4.119");}

^(pra|sam)[ ]jAnu[ ].* {uwwarapaxa_cgi_interface("NULL","jFu","प्रसंभ्याम्
जानुनोर्जुः 5.4.129");}
^UrXva[ ]jAnu[ ].*
    {uwwarapaxa_cgi_interface("NULL","jFu","ऊर्ध्वाद्विभाषा 5.4.130");}
^(xvi|anwar|pra|para|apa|sam|anu|ava|nis|nir|xus|xur|vi|A|ni|aXi|api|awi|su|uw|
aBi|prawi|pari|upa)[ ]ap[ ].*
    {uwwarapaxa_cgi_interface("NULL","Ip","द्व्यन्तरूपसर्गेभ्यः अपः ईत् 6.3.97");}
^{\prA}[ ]ciwi[ ].* {uwwarapaxa_cgi_interface("NULL","ciwI","चित्तेः
कपि 6.3.127");}
^(Sara|ikRu|plakRa|Amra|kArRya|Kaxira|pIyUkRA)[ ]vana[ ].*

```

```

{uwwarapaxa_cgi_interface("NULL","vaNa","प्रनिरन्तःशरेक्षुप्लक्षाम्रकार्ष्यखदिरपीयू
क्षाम्योऽसंज्ञायामपि 8.4.5");}
^(pra|nir|anwar)[ ]vana[ ].*

{uwwarapaxa_cgi_interface("NULL","vaNa","प्रनिरन्तःशरेक्षुप्लक्षाम्रकार्ष्यखदिरपीयू
क्षाम्योऽसंज्ञायामपि 8.4.5");}
^(puragA|miSrakA|siXrakA|sArikA|kotara|agra)[ ]vana[ ].*
    {uwwarapaxa_cgi_interface("NULL","vaNa"); strcpy(sUwrastr,"वनं
पुरगामिश्रकासिध्रकासारिकाकोटराग्रेभ्यः 8.4.4");}
^(sarva|saMkyAwa|pUrva|maXya)[ ]ahan[ ].*
    {uwwarapaxa_cgi_interface("NULL","ahna","अहोऽह एतेभ्यः 5.4.88");}
^{safKyA}[ ]ahan[ ].* { uwwarapaxa_cgi_interface("NULL","", "न संख्यादेः
समाहारे 5.4.89");}
^(puNya|eka)[ ]ahan[ ].* { uwwarapaxa_cgi_interface("NULL","", "उत्तमैकाभ्याम्
च 5.4.90");}
^{kumBAxi}[ ]pAxa[ ]bahuvrIhi {
    sprintf(ques,"किमयं समासः स्त्रियां वर्तते(Y/N)?");
    /* In case of bahuvrIhi should we not always ask the lifga? */
    uwwarapaxa_cgi_interface(ques,"paxI","कुम्भपदीषु च
5.4.139");
}
^{haswyAxi}[ ]pAxa[ ].*
    { uwwarapaxa_cgi_interface("NULL","pAxa","पादस्य लोपोऽहस्त्यादिभ्यः
5.4.138");}
^{(safKyA)|su)[ ]pAxa[ ].*
    { uwwarapaxa_cgi_interface("NULL","pAx","सङ्ख्यासुपूर्वस्य 5.4.140");}
^{prA}[ ]pAxa[ ].* { uwwarapaxa_cgi_interface("NULL","pAx","पादस्य
लोपोऽहस्त्यादिभ्यः 5.4.138");}
^su[ ]hqxaya[ ]bahuvrIhi {
    sprintf(ques,"किमनेन समासेन मित्रार्थः उच्यते(Y/N)?");
    uwwarapaxa_cgi_interface(ques,"hqw","सुहृद्बुद्धौ मित्रामित्रयोः
5.4.150");
}
^xur[ ]hqxaya[ ]bahuvrIhi {
    sprintf(ques,"किम् अनेन समासेन अमित्रार्थः उच्यते(Y/N)?");
    uwwarapaxa_cgi_interface(ques,"hqw","सुहृद्बुद्धौ मित्रामित्रयोः
5.4.150");
}
^agni[ ]swuw[ ].* { uwwarapaxa_cgi_interface("NULL","Rtuw","अग्नेः
स्तुत्स्तोमसोमाः 8.3.82");}
^agni[ ]swoma[ ].* { uwwarapaxa_cgi_interface("NULL","Rtoma","अग्नेः
स्तुत्स्तोमसोमाः 8.3.82");}
^agni[ ]soma[ ].* { uwwarapaxa_cgi_interface("NULL","Roma","अग्नेः

```

```

स्तुत्स्तोमसोमा: 8.3.82");}
^(jyowis|Ayus)[ ]swoma[ ].*
    { uwwarapaxa_cgi_interface("NULL","Rtoma","ज्योतिरायुषः स्तोमः
8.3.83");}
^.*r*a[ ]ahna[ ].*
    { uwwarapaxa_cgi_interface("NULL","ahNa","अहोऽदन्तात् 8.4.7");}
^.*    {uwwarapaxa_cgi_interface("NULL","", "");}

```

```

%%
char ques[LARGE];
char sUwrastr[LARGE];
char ustr[MEDIUMUTF];
char p1[MEDIUM];
char p2[MEDIUM];
char samAsAnwa[MEDIUM];
char samAsaprakAra[MEDIUM];
char encoding[MEDIUM];
int divid;

main (int argc, char *argv[]) {
char in[MEDIUM];
    strcpy(encoding,argv[1]);
    strcpy(p1,argv[2]);
    strcpy(p2,argv[3]);
    strcpy(samAsAnwa,argv[4]);
    strcpy(samAsaprakAra,argv[5]);
    divid = atoi(argv[6]);

    strcpy(in,p1);
    strcat(in," ");
    strcat(in,p2);
    strcat(in," ");
    strcat(in,samAsaprakAra);
    // printf("Calling uwwarapaxa lex = %s",in);
    call_uwwarapaxalex(in);
}
call_uwwarapaxalex(char *in){
    yy_scan_string(in);
    yylex();
}
yywrap()
{
}

```

```

uwwarapaxa_cgi_interface(char *condition, char *ustr, char *sstr){
    char conditionr[LARGE];

    if(strcmp(condition,"NULL")){
        printf("</tr></table><form name=\"f6\" method=\"post\" />");
    } else {
        printf("<td><form name=\"f6\" method=\"post\" />");
    }
    printf("<input type=\"hidden\" name=\"encodinguwwara\"
id=\"encodinguwwara\" value=\"%s\" />",encoding);
    printf("<input type=\"hidden\" name=\"p1uwwara\" id=\"p1uwwara\"
value=\"%s\" />",p1);
    printf("<input type=\"hidden\" name=\"p2uwwara\" id=\"p2uwwara\"
value=\"%s\" />",p2);
    printf("<input type=\"hidden\" name=\"samAsAnwauwwara\"
id=\"samAsAnwauwwara\" value=\"%s\" />",samAsAnwa);
    printf("<input type=\"hidden\" name=\"samAsaparakArauwwara\"
id=\"samAsaparakArauwwara\" value=\"%s\" />",samAsaparakAra);
    printf("<input type=\"hidden\" name=\"uwwarapaxastr\"
id=\"uwwarapaxastr\" value=\"%s\" />",ustr);
    printf("<input type=\"hidden\" name=\"sUwrastruwwara\"
id=\"sUwrastruwwara\" value=\"%s\" />",sstr);
    printf("<input type=\"hidden\" name=\"dividuwwara\" id=\"dividuwwara\"
value=\"#output%d\" />",divid);
    if(strcmp(condition,"NULL")){
        if(!strcmp(encoding,"RMN")){
            cnvrtutfd2r(condition,conditionr);
            printf("%s",condition);
        } else {
            printf("%s",condition);
        }
    }
    printf("<input type=\"submit\" name=\"ansuwwara\" value=\"Yes\"
id=\"ansuwwara\" onclick=\"return uwwarapaxayescgi()\" />");
    printf("<input type=\"submit\" name=\"ansuwwara\" value=\"No\"
id=\"ansuwwara\" onclick=\"return uwwarapaxanocgi()\" />");
    }
    else
        if (!strcmp(encoding,"RMN")) {
            printf("<input type=\"submit\" name=\"ansuwwara\"
value=\"Continue\" id=\"ansuwwara\" onclick=\"return
uwwarapaxaNullcgi()\" />");
        } else {
            printf("<input type=\"submit\" name=\"ansuwwara\"

```

```
value=\"Continue\" id=\"ansuwwara\" onclick=\"return
uwwarapaxaNullcgi()\"/>");
    }
    if(strcmp(condition,"NULL")){
        printf("</form>");
    } else {
        printf("</form></td></tr></table>");
    }
}
```

Padakarya.lex

```
%{
#include "struct.h"
extern char paxa[LARGE];
extern char sUwrastr[LARGE];
extern char samAsAnwa[MEDIUM];
}%

%x ppaxa
%x upaxa
%%

^ppaxa      {BEGIN ppaxa;}
^upaxa      {BEGIN upaxa;}

<ppaxa>[a-zA-Z]+n      {
    strcpy(paxa,yytext);
    if (strcmp(paxa,"ahan") && strcmp(paxa,"an")) {
        /* In ppaxa it is not necessary to chk samAsAnwaprawyaya.
Cmp with upaxa */
        paxa[yyleng-1] = '\0';
        /* ahan is taken care by the apavAda-sandhi rules
        ruwva-> uwva-> guNa";$ans2="roZsupi (8.2.69)
(prApwe) -> rUparAwri raWanwareRuruwvam vAcyam (vA 4847)-> haSi ca
(6.1.114)-> Ax guNaH (6.1.87)
        */
        strcpy(sUwrastr,"न लोपः प्रातिपदिकान्तस्य 8.2.7"); }
    else { strcpy(paxa,yytext); strcpy(sUwrastr,""); }
    BEGIN 0;
}

<ppaxa>.* {
    strcpy(paxa,yytext); strcpy(sUwrastr,"");
    BEGIN 0;
}

<upaxa>[a-zA-Z]+n      {
    strcpy(paxa,yytext);
    if ((samAsAnwa[0] == '\0') && strcmp(paxa,"ahan") &&
strcmp(paxa,"an") && strcmp(paxa,"rAjan")) {
        paxa[yyleng-1] = '\0';
        /* ahan is taken care by the apavAda-sandhi rules
        ruwva-> uwva-> guNa";$ans2="roZsupi (8.2.69)
(prApwe) -> rUparAwri raWanwareRuruwvam vAcyam (vA 4847)-> haSi ca
```

(6.1.114)-> Ax guNaH (6.1.87)

```
        */
        strcpy(sUwrastr,"न लोपः प्रातिपदिकान्तस्य 8.2.7"); }
    else { strcpy(paxa,yytext); strcpy(sUwrastr,""); }
    }
<upaxa>.*wvA    {
    strcpy(paxa,yytext); /* Temp fix to handle wvA -> lyap */
    paxa[yyleng-2] = '\0';
    strcat(paxa,"ya");
    paxa[yyleng] = '\0';
    }
<upaxa>.*    {
    strcpy(paxa,yytext); strcpy(sUwrastr,"");
    }
}
```

%%

```
char sUwrastr[LARGE];
char p1_utf[MEDIUMUTF];
char p2_utf[MEDIUMUTF];
char p1[MEDIUM];
char p2[MEDIUM];
char samAsAnwa[MEDIUM];
char samAsaprakAra[MEDIUM];
char paxa[LARGE];
char in[MEDIUM];
char putf[MEDIUMUTF]; /* utf takes 3 times space */
char encoding[MEDIUM];
```

```
int divid;
```

```
void do_samAsAnwaprawyaya_kArya();
void get_sandhied_word();
void do_tilopa();
void do_apaxAnwa_sandhi();
void get_iw();
```

```
main (int argc, char *argv[]) {
char sandhiwutf[MEDIUMUTF]; /* utf takes 3 times space */
char sandhiwutf[MEDIUMUTF];
char sandhiw[MEDIUM];
char putfr[MEDIUMUTF];
char paxautf[MEDIUMUTF]; /* utf takes 3 times space */
```

```

char paxautfr[MEDIUMUTF];
char sUwrastr2utf[LARGE];

strcpy(encoding,argv[1]);
strcpy(p1,argv[2]);
strcpy(p2,argv[3]);
strcpy(samAsAnwa,argv[4]);
strcpy(samAsaprakAra,argv[5]);
divid = atoi(argv[6]);

strcpy(in,"ppaxa");
strcat(in,p1);
call_paxalex(in);
if(strcmp(p1,paxa)){
    cnvrtwx2utf(p1, putf);
    cnvrtwx2utf(paxa, paxautf);
    if (!strcmp(encoding,"RMN")) {
        cnvrtutfd2r(putf,putfr);
        cnvrtutfd2r(paxautf,paxautfr);
        cnvrtutfd2r(sUwrastr,sUwrastr2utf);
        printf("<font color=\"blue\"> %s -> %s</font> <font color=\"red\"> %s
</font> <br />", putfr, paxautfr, sUwrastr2utf);
    } else {
        printf("<font color=\"blue\"> %s -> %s</font> <font color=\"red\"> %s
</font> <br />", putf, paxautf, sUwrastr);
    }
}
strcpy(p1_utf,paxa);

strcpy(in,"upaxa");
strcat(in,p2);
call_paxalex(in);
if(strcmp(p2,paxa)){
    cnvrtwx2utf(p2_utf, putf);
    cnvrtwx2utf(paxa, paxautf);
    if (!strcmp(encoding,"RMN")) {
        cnvrtutfd2r(putf,putfr);
        cnvrtutfd2r(paxautf,paxautfr);
        cnvrtutfd2r(sUwrastr,sUwrastr2utf);
        printf("<font color=\"blue\"> %s -> %s</font> <font color=\"red\"> %s
</font> <br />", putfr, paxautfr, sUwrastr2utf);
    } else {
        printf("<font color=\"blue\"> %s -> %s</font> <font color=\"red\"> %s

```

```

</font> <br />", putf, paxautf, sUwrastr);
    }
}
strcpy(p2_utf,paxa);
do_samAsAnwaprawyaya_kArya(p2_utf,samAsAnwa,samAsaparakAra,paxa);
if(strcmp(p2_utf,paxa)){
    cnvrtwx2utf(p2_utf, putf);
    cnvrtwx2utf(paxa, paxautf);
    if (!strcmp(encoding,"RMN")) {
        cnvrtutfd2r(putf,putfr);
        cnvrtutfd2r(paxautf,paxautfr);
        cnvrtutfd2r(sUwrastr,sUwrastr2utf);
        printf("<font color=\"blue\"> %s -> %s</font> <font color=\"red\"> %s
</font> <br />", putfr, paxautfr, sUwrastr2utf);
    } else {
        printf("<font color=\"blue\"> %s -> %s</font> <font color=\"red\"> %s
</font> <br />", putf, paxautf, sUwrastr);
    }
}
strcpy(p2_utf,paxa);

get_sandhied_word(p1_utf,paxa,sandhiw);
cnvrtwx2utf(sandhiw,sandhiwutf);
if (!strcmp(encoding,"RMN")) {
    cnvrtutfd2r(sandhiwutf,sandhiwutfr);
    printf("<font color=\"red\">%ssamastapadam:</font><font color=\"blue\">
%s</font>",sandhiwutfr);
} else {
    printf("<font color=\"red\">समस्तपदम्:</font><font color=\"blue\">
%s</font>",sandhiwutf);
}
}
call_paxalex(char *in){
    yy_scan_string(in);
    yylex();
}
yywrap()
{
}

void do_samAsAnwaprawyaya_kArya(char *inp, char *sp, char *prakAra, char
*out) {
char sp_noiw[MEDIUM];

```

```

char paxa[MEDIUM];

/* It is assumed that the samAsAnwa prawyaya are waxXiwa prawyayas */
if(strcmp(sp,"")){
    get_iw(sp, sp_noiw);
    // printf("Within do_samAsAnwaprawyaya_kArya");
    // printf("sp = %s samAsaprakAra = %s inp = %s",sp,samAsaprakAra,inp);
    // printf("inp -1 = %c inp -2 = %c",inp[strlen(inp)-1],inp[strlen(inp)-2]);

    if((!strcmp(sp,"tac") || !strcmp(sp,"Ra")) &&
        (!strcmp(samAsaprakAra,"avyayIBAvH") ||
         !strcmp(samAsaprakAra,"wawpuruRa") || !
strcmp(samAsaprakAra,"bahuvrIhi")) &&
        (inp[strlen(inp)-1]=='n') &&
        (inp[strlen(inp)-2]=='a')) {
        // printf("calling do_tilopa");
        do_tilopa(inp,paxa);
        strcpy(inp,paxa);
    }
    if(ajAxi(sp_noiw) &&
        ((inp[strlen(inp)-1] == 'i') ||
         (inp[strlen(inp)-1] == 'a'))
        ) {
        inp[strlen(inp)-1] = '\0';
    }

    do_apaxAnwa_sandhi(inp,sp_noiw,out);
} else {strcpy(out,inp);}
}

void do_tilopa(char *inp, char *out){
int len;
len = strlen(inp);
while(!isvowel(inp[len])) len--;
inp[len] = '\0';
strcpy(out,inp);
}

void do_apaxAnwa_sandhi(char *in1, char *in2, char*out){
int len1,len2;
strcpy(out,in1);
strcat(out,in2);
len1 = strlen(in1);

```

```

len2 = strlen(in2);
out[len1+len2] = '\0';
}

```

```

int ajAxi(char *p){
    char ac[14],i;
    strcpy(ac,"aAiIuUqQLeEoO");
    if(index(ac,p[0])) { return 1;} else {return 0;}
}

```

```

void get_iw(char *sp, char *sp_noiw) {
    strcpy(sp_noiw,sp);
    if(!strcmp(sp,"tac")) { strcpy(sp_noiw,"a");}
    if(!strcmp(sp,"ac")) { strcpy(sp_noiw,"a");}
    if(!strcmp(sp,"Rac")) { strcpy(sp_noiw,"a");}
    if(!strcmp(sp,"Ra")) { strcpy(sp_noiw,"a");}
    if(!strcmp(sp,"ap")) { strcpy(sp_noiw,"a");}
    if(!strcmp(sp,"asic")) { strcpy(sp_noiw,"as");}
    if(!strcmp(sp,"anic")) { strcpy(sp_noiw,"an");}
    if(!strcmp(sp,"ic")) { strcpy(sp_noiw,"i");}
    if(!strcmp(sp,"anaf")) { strcpy(sp_noiw,"an");}
    if(!strcmp(sp,"kap")) { strcpy(sp_noiw,"ka");}
}

```

```

int isvowel(int ch){
    switch (ch){
        case 'a':
        case 'A':
        case 'i':
        case 'I':
        case 'u':
        case 'U':
        case 'q':
        case 'Q':
        case 'L':
        case 'e':
        case 'E':
        case 'o':
        case 'O': return 1; break;
        default: return 0;
    }
}

```

SCREENSHOTS

समस्तपदव्युत्पादिका : A compound stem generator

Department of Sanskrit Studies, University of Hyderabad.

अधः प्रदत्तम् अलौकिकविग्रहवाक्यं चिन्तन्तु

राजन्+ङस् पुरुष+सु (2.2.8)
महत्+सु राजन्+सु (2.1.61)
कुङ्गलाग्र+सु दन्त+सु (2.2.24)
नील+सु उत्पल+सु (2.1.57)
ऊढ+सु रथ+सु (2.2.24)
एक+टा परि (2.1.10)
कु अन्न+सु (2.2.18)
कु रथ+सु (2.2.18)
कु वद+सु (2.2.18)
हृदय+ङस् रोग+सु (2.2.8)
हृदय+ङस् लास+सु (2.2.8)
हरि+सु हर+सु (2.2.29)
हल+ङि त्रिपदिका+सु (2.1.44)
हस्त+ङि बन्ध+सु (2.1.41)
हस्तिन्+ङस् वर्चस्+सु (2.2.8)
न कर्तुम् (2.2.6)
अकेशा+सु भार्या+सु (2.2.24)
अक्ष+ङस् धुर+सु (2.2.8)
अक्ष+टा परि (2.1.10)
अक्ष+सुप् शौण्ड+सु (2.1.40)

submit

राजन्+ङस् पुरुष+सु समासप्रकारः तत्पुरुष षष्ठी 2.2.8

Continue

राजन्+ङस् पुरुष+सु समासान्तप्रत्ययः --

राजन्+ङस् पुरुष+सु प्रातिपदिकसंज्ञा कृतक्षितसमासाश्च 1.2.46

Continue

राजन्+ङस् पुरुष+सु उपसर्जनसंज्ञा : राजन् प्रथमानिर्दिष्टं समास उपसर्जनम् 1.2.43

राजन्+ङस् पुरुष+सु पूर्वनिपातः उपसर्जनं पूर्वम् 2.2.30

किम् अनेन समासेन आक्रोशार्थः गम्यते(Y/N)?

Yes

No

राजन्+[ङस्] पुरुष+[सु] सुपो धातुप्रातिपदिकयोः 2.4.71

Continue

Continue

राजन्-> राज न लोपः प्रातिपदिकान्तस्य 8.2.7

समस्तपदम्: राजपुरुष

समस्तपदव्युत्पादिका : A compound stem generator

Department of Sanskrit Studies, University of Hyderabad.

अधः प्रदत्तम् अलौकिकविग्रहवाक्यं विन्यन्तु

राजन्+ङस् पुरुष+सु (2.2.8)
महत्+सु राजन्+सु (2.1.61)
कुङ्मलाग्र+सु दन्त+सु (2.2.24)
नील+सु उत्पल+सु (2.1.57)
ऊढ+सु स्थ+सु (2.2.24)
एक+टा परि (2.1.10)
कु अन्न+सु (2.2.18)
कु रथ+सु (2.2.18)
कु वद+सु (2.2.18)
हृदय+ङस् रोग+सु (2.2.8)
हृदय+ङस् लास+सु (2.2.8)
हरि+सु हर+सु (2.2.29)
हल+ङि विपदिका+सु (2.1.44)
हस्त+ङि बन्ध+सु (2.1.41)
हस्तिन्+ङस् वर्चस्+सु (2.2.8)
न कर्तुम् (2.2.6)
अकेशा+सु भार्या+सु (2.2.24)
अक्ष+ङस् क्षुर+सु (2.2.8)
अक्ष+टा परि (2.1.10)
अक्ष+सुप् शौण्ड+सु (2.1.40)

submit

महत्+सु राजन्+सु समासप्रकारः तत्पुरुष सन्महत्परमौत्तमोत्कृष्टः पूज्यमानैः 2.1.61 [Continue](#)

महत्+सु राजन्+सु समासान्तप्रत्ययः टच् राजाहस्सखिभ्यष्टच् 5.4.91

महत्+सु राजन्+सु टच् प्रातिपदिकसंज्ञा कृतद्धितसमासाश्च 1.2.46

अनेन समासेन पूज्यमानार्थः गम्यते(Y/N)? [Yes](#) [No](#)

महत्+सु राजन्+सु उपसर्जनसंज्ञा : महत् प्रथमानिर्दिष्टं समास उपसर्जनम् 1.2.43

महत्+सु राजन्+सु पूर्वनिपातः उपसर्जनं पूर्वम् 2.2.30 [Continue](#)

महत् +[सु] राजन् +[सु] टच् सुपो धातुप्रातिपदिकयोः 2.4.71 [Continue](#)

महत् -> महा आन्महतः समानाधिकरणजातीययोः 6.3.46

[Continue](#)

राज् -> राज

समस्तपदम्: महाराज

समस्तपदव्युत्पादिका : A compound stem generator

Department of Sanskrit Studies, University of Hyderabad.

अधः प्रदत्तम् अलौकिकविग्रहवाक्यं चिन्तयन्तु

राजन्+ङस् पुरुष+सु (2.2.8)
महत्+सु राजन्+सु (2.1.61)
कुङ्मलाग्र+सु दन्त+सु (2.2.24)
नील+सु उत्पल+सु (2.1.57)
ऊढ+सु रथ+सु (2.2.24)
एक+टा परि (2.1.10)
कु अन्न+सु (2.2.18)
कु रथ+सु (2.2.18)
कु वद+सु (2.2.18)
हृदय+ङस् रोग+सु (2.2.8)
हृदय+ङस् लास+सु (2.2.8)
हरि+सु हन्+सु (2.2.29)
हल+ङि विपदिका+सु (2.1.44)
हस्त+ङि बन्ध+सु (2.1.41)
हस्तिन्+ङस् वर्चस्+सु (2.2.8)
न कर्तुम् (2.2.6)
अकेशा+सु भार्या+सु (2.2.24)
अक्ष+ङस् धुर+सु (2.2.8)
अक्ष+टा परि (2.1.10)
अक्ष+सुप् शौण्ड+सु (2.1.40)

submit

न कर्तुम् समासप्रकारः तत्पुरुष नञ् 2.2.6

Continue

न कर्तुम् समासान्तप्रत्ययः --

न कर्तुम् प्रातिपदिकसंज्ञा कृतद्धितसमासाश्च 1.2.46

Continue

न कर्तुम् उपसर्जनसंज्ञा : न प्रथमानिर्दिष्ट समास उपसर्जनम् 1.2.43

न कर्तुम् पूर्वनिपातः उपसर्जनं पूर्वम् 2.2.30

Continue

न + [] कर्तुम् + [] सुपो धातुप्रातिपदिकयोः 2.4.71

Continue

न -> अ नलोपो नञः 6.3.73

Continue

समस्तपदम्: अकर्तुम्

समस्तपदव्युत्पादिका : A compound stem generator

Department of Sanskrit Studies, University of Hyderabad.

अधः प्रदत्तम् अलौकिकविग्रहवाक्यं चिन्तयतु

अपरा+सु समा+सु (2.1.17)
अपर+सु अध्यापक+सु (2.1.58)
विष्णु+ङस् अप (2.1.12)
अपूप+ङस् अर्ध+सु (2.2.8)
अप्+सु चर (2.2.19)
अप्+सु ज (2.2.19)
अप्+सु मति+सु (2.2.24)
अप्+सुप् योनि+सु (2.2.24)
न ब्राह्मन+सु (2.2.6)
अग्नि+अम् अभि (2.1.14)
अभ्याश+ङसि आगत+सु (2.1.39)
अमृत+सु अश्मन्+सु (2.1.57)
अयस्+सु दन्त+अस् (2.2.24)
अरण्य+अम् गत+सु (2.1.24)
अरण्य+ङि तिलक+अस् (2.1.44)
न राजन्+सु (2.2.6)
शिरस्+ङि अरुप्+सु (2.2.24)
अरोक्+जस् दन्त+जस् (2.2.24)
अर्क+सु अश्वमेध+सु (2.2.29)
अर्थ+ङस् गौरव+सु (2.2.8)

submit

अग्नि+अम् अभि समासप्रकारः अव्ययीभावः लक्षणैः अभिप्रती अभिमुख्ये 2.1.14 [Continue](#)

अग्नि+अम् अभि समासान्तप्रत्ययः --

अग्नि+अम् अभि प्रातिपदिकसंज्ञा कृतद्वितसमासाश्च 1.2.46

अभिप्रती अभिमुख्यार्थे वर्तते एवम् इदम् पदम् अग्नि चिह्नवाचकम् वा(Y/N)? [Yes](#) [No](#)

अग्नि+अम् अभि उपसर्जनसंज्ञा : अभि प्रथमानिर्दिष्ट समास उपसर्जनम् 1.2.43

अभि अग्नि+अम् पूर्वनिपातः उपसर्जनं पूर्वम् 2.2.30

[Continue](#)

अभि + [] अग्नि + [अम्] सुपो धातुप्रातिपदिकयोः 2.4.71

[Continue](#)

[Continue](#)

समस्तपदम्: अभ्यग्नि

समस्तपदव्युत्पादिका : A compound stem generator

Department of Sanskrit Studies, University of Hyderabad.

अधः प्रदत्तम् अलौकिकविग्रहयावत् चिन्तितु

- सु सामन्+सु (2.2.24)
- सु सक्थि+औ (2.2.24)
- सु सक्थि+सु (2.2.24)
- सु हलि+सु (2.2.24)
- सु हलि+सु (2.2.24)
- सु हृदय+सु (2.2.24)
- सु हृदय+सु (2.2.24)
- समान+सु उदर+सु (2.1.58)
- स्नात+सु अनुलिप्त+सु (2.1.49)
- हरि+सु चन्द्र+सु (2.1.55)
- हरि+सु सक्थि+सु (2.2.8)
- हरि+सु हर+सु (2.2.29)
- हल+ङि विपदिका+सु (2.1.44)
- हस्त+ङि बन्ध+सु (2.1.41)
- हस्तिन्+ङस् वर्चस्+सु (2.2.8)
- हृदय+ङस् शोक+सु (2.2.8)
- हृदय+ङस् रोग+सु (2.2.8)
- हृदय+ङस् लेख+सु (2.2.8)
- हृदय+ङस् लास+सु (2.2.8)
- हृदय+ङस् लेख (2.2.19)

submit

हृदय+ङस् लेख+सु समासप्रकारः तत्पुरुष षष्ठी 2.2.8 [Continue](#)

हृदय+ङस् लेख+सु समासान्तप्रत्ययः --

हृदय+ङस् लेख+सु प्रातिपदिकसंज्ञा कृतद्धितसमासाश्च 1.2.46 [Continue](#)

हृदय+ङस् लेख+सु उपसर्जनसंज्ञा : हृदय प्रथमानिर्दिष्टं समास उपसर्जनम् 1.2.43

हृदय+ङस् लेख+सु पूर्वनिपातः उपसर्जनं पूर्वम् 2.2.30

किम् अनेन समासेन आक्रोशार्थः गम्यते(Y/N)? [Yes](#) [No](#)

हृदय +[ङस्] लेख +[सु] सुपो धातुप्रातिपदिकयोः 2.4.71 [Continue](#)

हृदय -> हृत् हृदयस्य हृत्लेखयदणलासेषु 6.3.50

[Continue](#)

समस्तपदम्: हृत्लेख

समस्तपदव्युत्पादिका : A compound stem generator

Department of Sanskrit Studies, University of Hyderabad.

अधः प्रदत्तम् अलौकिकविग्रहवाक्यं चिन्तयन्तु

गोपा+ङि अधि (2.1.6)
हरि+ङि अधि (2.1.6)
आत्मन्+ङि अधि (2.1.6)
न अश्च+सु (2.2.6)
ग्राम+ङस् अनु (2.1.16)
गो+ङस् अनु (2.1.16)
गिरि+ङस् अनु (2.1.6)
गिरि+ङस् अनु (2.1.6)
गो+ङस् अनु (2.1.6)
ज्येष्ठ+ङस् अनु (2.1.6)
न उपलब्धि+सु (2.2.6)
अनु रहस्+सु (2.2.18)
रूप+ङस् अनु (2.1.6)
वन+ङस् अनु (2.1.15)
विष्णु+ङस् अनु (2.1.6)
गिरि+ङि अन्तर (2.1.6)
अन्तिक+ङसि आगत+सु (2.1.39)
अन्ध+सु तमस्+सु (2.1.57)
अन्य+ङस् कारक+सु (2.2.8)
अन्य+ङे अर्थ+सु (2.1.36)

submit

न उपलब्धि+सु समासप्रकारः तत्पुरुष नञ् 2.2.6	Continue
न उपलब्धि+सु समासान्तप्रत्ययः -- नञस्तत्पुरुषात् 5.4.71	
न उपलब्धि+सु प्रातिपदिकसंज्ञा कृतद्धितसमासाश्च 1.2.46	Continue
न उपलब्धि+सु उपसर्जनसंज्ञा : न प्रथमानिर्दिष्ट समास उपसर्जनम् 1.2.43	
न उपलब्धि+सु पूर्वनिपातः उपसर्जनं पूर्वम् 2.2.30	Continue
न +[] उपलब्धि +[सु] सुपो धातुप्रातिपदिकयोः 2.4.71	Continue
न -> अन् तस्मान्नुडचि 6.3.74	
Continue	
समस्तपदम्: अनुपलब्धि	

Bibliography

Bhaṭṭojidīkṣitaḥ. *Vaiyākaraṇasiddhantakaumudī*. Motilal Banarsidass, New Delhi, 2001.

V. V. Bhandare. Structural and Semantic Aspects of the Dvandva Compound. *Annals of the Bhandarkar Oriental Research Institute*, 76(1-4): 89–96, 1995.

Akshar Bharati and Amba Kulkarni. Information Coding in a Language: Some Insights from Pāṇinian Grammar. *Dhimahi, Journal of Chinmaya International Foundation Shodha Sansthan*, 80(1):77–91, 2010.

Saroja Bhate and Subhash Kak. Pāṇini’s Grammar and Computer Science. *Annals of the Bhandarkar Oriental Research Institute*, 1991-1992(1/4): 79–94, 1992.

George Cardona. Studies in Indian Grammarians I: The Method of Description Reflected in the śivasūtras. In *Transactions of the American Philosophical Society*, pages 3–48. JSTOR, 1969.

George Cardona. Pāṇini: His Work and its Traditions vol 1. In *Background and Introduction*. 2nd ed. Motilal Banarsidass, 1997.

- George Cardona. On the Structure of Pāṇini's System. In *Sanskrit Computational Linguistics, First and Second International Symposia, Rocquencourt, France*, pages 1–32. Springer, 2009.
- Ashwini Deo. Derivational Morphology in Inheritance-based Lexica: Insights from Pāṇini. In *Lingua*, volume 117.1, pages 175–201. Elsevier, 2007.
- Brendan S. Gillon. Exocentric Compounds in Classical Sanskrit. In Amba Kulkarni and Gérard Huet, editors, *Sanskrit Computational Linguistics*, 2007.
- Brendan S. Gillon. Tagging Classical Sanskrit Compounds. In Amba Kulkarni and Gérard Huet, editors, *Sanskrit Computational Linguistics 3*, pages 98–105. Springer-Verlag LNAI 5406, 2009.
- Pawan Goyal and Gérard Huet. Completeness Analysis of a Sanskrit Reader. In *Proceedings, 5th International Symposium on Sanskrit Computational Linguistics*. D. K. Printworld(P) Ltd, 2013.
- Pawan Goyal, Amba Kulkarni, and Laxmidhar Behera. Computer Simulation of Aṣṭādhyāyī: Some Insights. In *Sanskrit Computational Linguistics, First and Second International Symposia, Rocquencourt, France*, pages 139–161. Springer, 2009.
- F. Grimal, V. Venkataraja Sarma, and S. Lakshminarasimham. *Pāṇinīyavyākaraṇodāharaṇakośaḥ*. Prentice-Hall of India, New Delhi, 1995.
- Rodney Huddleston and Geoffrey K. Pullum. *The Cambridge Grammar of the English Language*. Cambridge University Press, 2002.

- Gérard Huet. Design of a Lexical Database for Sanskrit. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*, pages 8–14. Association for Computational Linguistics, 2004.
- Malcolm D. Hyman. From Pāṇinian Sandhi to Finite State Calculus. In *Sanskrit Computational Linguistics, First and Second International Symposia, Rocquencourt, France*, pages 253–265. Springer, 2009.
- Girish Nath Jha and Sudhir K. Mishra. Semantic Processing in Pāṇinis Kāraka System. In *Sanskrit Computational Linguistics, First and Second International Symposia, Rocquencourt, France*, pages 239–252. Springer, 2009.
- S. D. Joshi. *Patañjali's Vyākaraṇa - Mahābhāṣya Samarthāhnika (P. 2.1.1)*. Centre of Advanced Study in Sanskrit, University of Poona, Poona, 1968.
- S. D. Joshi. Background of Aṣṭādhyāyī. pages 1–5, 2009.
- S. D. Joshi and J. A. F. Roodbergen. On Siddha, Asiddha and Sthānivat. In *Annals of the Bhandarkar Oriental Research Institute*, volume 68, pages 541–549. Bhandarkar Oriental Research Institute, 1987.
- S. D. Joshi and J. A. F. Roodbergen. On Kāraka, Vibhakti and Samāsa. *Annals of the Bhandarkar Oriental Research Institute*, 80(1-4):95–112, 1999.
- Kaṇḍabhaṭṭaḥ. *Vaiyākaraṇabhūṣaṇasāraḥ*. Chaukamba Sanskrit Pratisthan, 2005.
- Kiparsky. Pāṇini is slick, but he isn't mean. *Nagoya Studies in Indian Culture and Buddhism, Sambhāṣā*, 26(1-28), 2007.

- András Kornai. *Mathematical Linguistics*. Springer-Verlag London Ltd, 2008.
- Amba Kulkarni and Anil Kumar. Statistical Constituency Parser for Sanskrit Compounds. In *Proceedings of ICON 2011*. Macmillan Advanced Research Series, Macmillan Publishers India Ltd., 2011.
- Malhar Kulkarni. Phonological Overgeneration in Pāṇinian System. In *Sanskrit Computational Linguistics, First and Second International Symposia, Rocquencourt, France*, pages 306–319. Springer, 2009.
- Anil Kumar. *An Automatic Sanskrit Compound Processor*. University of Hyderabad, 2012.
- Anil Kumar, Vipul Mittal, and Amba Kulkarni. Sanskrit Compound Processor. In G N Jha, editor, *Proceedings of the 4th International Sanskrit Computational Linguistics Symposium*. Springer-Verlag LNAI 6465, 2010.
- Mahavir. Treatment of Samāsa in Pāṇini. *Annals of the Bhandarkar Oriental Research Institute*, 67(1-4):147–158, 1986.
- Anand Mishra. Simulating the Pāṇinian System of Sanskrit Grammar. In *Sanskrit Computational Linguistics, First and Second International Symposia, Rocquencourt, France*, pages 127–138. Springer, 2009.
- Anand Mishra. Modelling Aṣṭādhyāyī: An Approach Based on the Methodology of Ancillary Disciplines (vedāṅga). In *Sanskrit Computational Linguistics, Fourth International Symposium, Delhi, India*, pages 239–258. Springer, 2010.

- M. Srimannarayana Murti. *Sanskrit Compounds. A Philosophical Study*. The Chowkamba Sanskrit series Office, Varanasi, India, 1974.
- Sivaja S. Nair. *The Knowledge Structure in Amarakośa*. University of Hyderabad, 2011.
- Sivaja S. Nair, Shrinivasa Varakhedi, and V. Sheeba. Extended Nyāya-Vaiśeṣika Ontology as Applied to Amarakośa Knowledgenet. In Malhar Kulkarni and Chaitali Dangarikar, editors, *Recent Researches in Sanskrit Computational Linguistics Fifth International Symposium Proceedings*, pages 223–242. D. K Printworld (P) Ltd, New Delhi, 2013.
- Bhagyalata Pataskar. Some Observations about the Compound Structure of Aṣṭādhyāyī. *Annals of the Bhandarkar Oriental Research Institute*, 77 (1-4):121–131, 1996.
- Gerald Penn and Paul Kiparsky. On Pāṇini and the Generative Capacity of Contextualized Replacement Systems. In *Proceedings of COLING 2012: Posters*, pages 943–950, 2012.
- Wiebke Petersen. A Mathematical Analysis of Pāṇini’s śivasūtras. 13: 471489, 2004.
- Rama. Inheritance-based Lexica: Insights from Pāṇini. In *Lingua 117.1*, 2007.
- K. V. Ramakrishnamacharya. *Bhūṣaṇasāratattvaparakāśikā*. Rashtriya Sanskrit Vidyapeetam, Tirupati, 2010.
- Pavankumar Satuluri and Amba Kulkarni. Extra Linguistic Information needed for Automatic Generation of Sanskrit Compounds: A Study.

- In *The recent developments in Sanskrit Computational Linguistics*, at SALA-30, Hyderabad, 2014.
- Peter Scharf. Modeling Pāṇinian Grammar. In *Sanskrit Computational Linguistics, First and Second International Symposia, Rocquencourt, France*, pages 95–126. Springer, 2009.
- Peter Scharf and Malcolm Hyman. *Linguistic Issues in Encoding Sanskrit*. Motilal Banarsidass, Delhi, 2009.
- Peter Scharf, Pawan Goyal, Anuja Ajotikar, and Tanuja Ajotikar. Voice, Preverb, and Transitivity Restrictions in Sanskrit Verb use. In *Sanskrit Syntax, Selected papers presented at the seminar on sanskrit syntax and discourse structures (forthcoming)*, 2015.
- Peter M Scharf. Rule-Blocking and Forward-Looking Conditions in the Computational Modelling of Pāṇinian Derivation. In *Sanskrit Computational Linguistics, Fourth International Symposium. New Delhi*, 2010a.
- Peter M Scharf. Rule Selection in the Aṣṭādhyāyī, or Is Pāṇinis Grammar Mechanistic? In *Studies in Sanskrit Grammars: Proceedings of the Vyakarana Section of the 14th World Sanskrit Conference*, 2010b.
- Fritz Staal. Context-Sensitive Rules in Pāṇini. 1(1):63–72, 1965.
- Fritz Staal. Encyclopedia of Language and Linguistics. volume 6, page 2917, 1994.
- Sridhar Subbanna and Shrinivasa Varakhedi. Computational Structure of the Aṣṭādhyāyī and Conflict Resolution Techniques. In *Sanskrit Computational Linguistics, Third International Symposium, Hyderabad, India*, pages 56–65. Springer, 2009.

Sridhar Subbanna and Shrinivasa Varakhedi. Asiddhatva Principle in Computational Model of Aṣṭādhyāyī. In *Sanskrit Computational Linguistics, Fourth International Symposium, Delhi, India*, pages 231–238. Springer, 2010.