Parser for Simple Sanskrit Sentences
based on Pāṇinian Grammar Formalism

Master of Philosophy
in
Shabdabodha Systems and Language Technologies

By
N. Shailaja

Department of Sanskrit Studies
School of Humanities
University of Hyderabad
Hyderabad 500046
June 2009

Declaration

I hereby declare that the work embodied in this dissertation entitled **Parser for Simple Sanskrit Sentences based on Pāṇinian Grammar formalism** is carried out by me under the supervision of Amba Kulkarni, 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.

N Shailaja

08HSLS03

Date:

Place: Hyderabad

Department of Sanskrit Studies
University of Hyderabad

CERTIFICATE

This is to certify that **N Shailaja** has carried out the research-work embodied in the present dissertation entittled **" Parser for Simple Sanskrit Sentences based on Pāṇinian Grammar Formalism "** at the University of Hyderabad. The dissertation represents her independent work and has not been submitted for any research degree of this university or any other university.

Amba Kulkarni
Supervisor

Amba Kulkarni
Head
Department of Sanskrit Studies

Mohan G Ramanan
Dean
School of Humanities
University of Hyderabad

ACKNOWLEDGMENTS

At the outset, I would like to thank to God who gave me lot of energy to complete this work and then I would like to thank to ***Acharya Anand Prakash*** and ***Mrs B. Neeraja*** for their precious and valuable guidance and they enlightened me from the darkness.

The single most important person that must be acknowledged here, without whom this dissertation would not have been there, is ***Mrs. Amba Kulkarni*** who not only suggested this topic but also came through in the process of the research.

I am grateful indebted to my supervisor ***Mrs. Amba Kulkarni*** for her valuable guidance. She have been extremely cordial and co-operative during the research and I am thankful to her support and encouragement.

I am sincerely thankful to ***Prof. K. V. Ramakrishnamacharya & Prof. Malhar kulkarni*** who has been cleared my doubts.

I am heartly thankful to ***Prof. Vineet Chaitanya (IIIT), Dr. Dipti Mishra (IIIT)*** and ***Dr. Laxmi Bai (IIIT)*** for their support and encouragement.

I am thankful to my seniors ***Mr. Anil Kumar Gupta & Ms. Shivaja S.Nair*** for helping me a lot.

I am very much thankful to my classmates ***Mr. Abart, Mr. Shiva & Ms. Ramya sai*** for their helping in many ways.

I express my gratitude towards my parents for their love and affection and for their continuous encouragement and moral support. I express my special thanks to my husband ***Mr. Lakshman*** who gave me encouragement and moral support to finish this work.

I wish to thank them too whom I could not list out but directly or indirectly they helped me a lot.

N.Shailaja

Contents

Chapter 1

Introduction

व्याख्याकारं वररुचिं भाष्यकारं पतञ्जलिम्।
पाणिनिं सूत्रकारं च प्रणतोऽस्मि मुनित्रयम्॥

India has rich heritage in linguistic studies. Out of the six वेदाङ्ग-s (fields of studies necessary to study the vedas) viz. शिक्षा, व्याकरण, छन्दस्, निरुक्त, ज्योतिष and कल्प, the first four are concerned with language studies. शिक्षा deals with pronnciation, व्याकरण with grammatical aspects, छन्दस् with prosody and निरुक्त with etymology. Among all these the importance of व्याकरण is long recognised and is evident from the enormous literature on व्याकरण. It has a major role to play in understanding how a language communicates thoughts from one human being to the other.

पाणिनि in around 500 BC presented a concise and almost exhaustive descriptive coverage of the then prevalent Sanskrit language. This grammar is in the form of सूत्रs – around 4000 divided into 8 chapters of 4 sections each. "पाणिनि's grammar is universally admired for its insightful analysis of Sanskrit"(Kiparsky, 2002).

कात्यायन supplemented the work of पाणिनि with वार्तिकs. Later on पतञ्जलि wrote a detailed commentary on these two works ruling out the necessity of some of the वार्तिकs. This rich tradition continued further with major contributions from भर्तृहरि in the field of Language philosophy, and later by नागेश भट्ट, कौण्ड भट्ट, etc.

The studies in grammar further led to three important schools – व्याकरण, मीमांसा and न्याय. These are popularly known as पद, वाक्य and प्रमाण शास्त्र respectively, indicating the nature of the topics they deal with.

It is evident from the scientific literature that Sanskrit was "Lingua franca" of scientific discussions in India. But that material is today not accessible to the scientists who do not have enough exposure to Sanskrit. With the help of language technology, today it is possible to develop language processing

tools to provide access to these Sanskrit texts.

With the advent of computers, and availability of extensive Sanskrit text in electronic form, the vast Sanskrit literature is now available to the scholars easily. But still, those who do not know Sanskrit well, can not understand it. The advantages in the field of Machine Translation, and the availability of पाणिनि's अष्टाध्यायी can lead to the development of appropriate tools for accessing Sanskrit texts.

The aim of this thesis is to build a parser for simple Sanskrit sentences. This parser will analyse simple Sanskrit sentences and assign कारक roles to various nouns in a sentence. The parser will be based on पाणिनि's grammar.

This being an interdisciplinary work, in the next chapter, we give a brief background of the Panini's sutra related to विभक्ति and करक. The third chapter illustrate the process of शाब्दबोध theories the अन्वय of given words. The fourth chapter gives a breif summary of earlier efforts in this area. The fifth chapter introduces a computational environment - an expert system for programming viz CLIPS(C Language Integrated Production System), which is being used for developing the parser. Sixth chapter discusses the actual

implementation explaning sample rules. We cunclude with the directions for future work. In the seventh chapter we discuss the problem cases, and discuss the possible ways of solving them. The simple parser has been tested on a set of around1 130 sentences from introductory grammer books. The evaluation report is in the eighth chapter. The final chapter lists the future tasks and resources that are needed to build a realistic parser for Sanskrit.

Chapter 2

kAraka and vibhakti

Sanskrit is a highly inflectional language. Every word, be it a noun or a verb has to be inflected before it is used in a sentence. An inflected word is termed पद [1] and अपदम् न प्रयुञ्जीत is the verdict of a grammarian (one must not use a word, unless it is inflected). There are seven विभक्तिs (case-affixes) which make a crude form viz. प्रातिपदिक usable. These are called कारक विभक्तिs if they denote a relation between a noun and a verb. When these विभक्तिs are governed by on उपपद, they are known as upapada विभक्तिs. In the following sections, we give a brief note on both of these. But before that, let us first understand what a कारक is.

[1] सुप्तिङन्तम् पदम् (1.4.14)

2.1 कारक

कारक is the name given to the relation between a noun and a verb in a sentence. The literal meaning of the word is: any factor which contributes to the accomplishment of an action[2].

करोति = क्रियां निर्वर्तयति इति कारकम्।[3]

यत् क्रियोत्पत्तौ कारणं तदेव कारकम्।

महाभाष्ये उक्तम्- "कारकमिति महती संज्ञा क्रियते ... तत्र महत्याः संज्ञायाः करणे एतत् प्रयोजनं यथा अन्वर्थसंज्ञा विज्ञायेत्, करोतीति कारकम्". In वाक्यपदीय also भर्तृहरि states

स्वाश्रये समवेतानाम् तद्वदेवाश्रयान्तरे।

क्रियाया अभिनिष्पत्तौ सामर्थ्यं साधनम् विदुः॥

Here **साधन** is a synonym of **कारक**.

The relation of a noun to another noun in a sentence is not a **कारक**. A **कारक** must be related with a verb: **क्रियान्वयित्वम् कारकत्वम्**. There are six **कारक**s: **कर्त्ता**, **कर्म**, **करण**, **सम्प्रदान**,**अपादान** and **अधिकरण**. Appendix I lists all the **सूत्र**s that discribe these **कारक**s . All these **सूत्र**s come under the **कारके**. These

[2]कारकाणि हि स्वस्वव्यापारेण अवान्तर क्रियाद्वारेण वा केनापिरूपेण प्रधान क्रियोत्पतौ सहायकानि भवन्ति।

[3]क्रियानिष्पादकत्वं कारकत्वम् अर्थात् क्रियोत्पादकरूपार्थयुक्तत्वं कारकत्वमिति

सूत्रs describe the semantic meaning of the कारकs. पाणिनि further in the first chapter fourth पाद gives सूत्रs which describe the विभक्तिs used for realisation of various कारक relations.

पाणिनि starts his कारक section with the अधिकार सूत्र "कारके" (1.4.23). We describe in brief, each of the कारकs, as described by the पाणिनि's rule, followed by the rule governing the vibhakti of that कारक, followed by an example. We do not discuss the exceptional cases.

Sanskrit has three voices – कर्तृ, कर्म and भाव. In case of कर्तृवाच्य (roughly active voice), the कर्त्ता is said to be expressed through the verbal suffix. If a verb is सकर्मक (roughly transitive), it is the कर्म which is expressed through the verbal suffix, and finally in case of अकर्मक (roughly intransitive) verbs, what is expressed through the verbal suffix is the action. Now let us see more about these 6 कारकs.

2.1.1 कर्त्ता

कर्त्ता is defined as स्वतन्त्रः कर्त्ता (1.4.54). The most independent participant in the action is termed as karttA. It is the speaker's intention (विवक्षा) which assigns different roles to the participants of an action. Speaker 'expresses' the situation/event through the words. While wording his thoughts/expressions, he is free to assign the roles to various participants according to his intention. So for example, consider the situation: There are 4 participants: Devadatta,

स्थाली(vessel), odana(rice), काष्ठ(wood pieces) and finally the action of cooking(पाचन क्रिया). If the speaker is simply narrating an incidence that a cook by name Devadadda is cooking the rice in a vessel, he would say

देवदत्तः स्थाल्याम् ओदनम् पचति।

But if his intention is to express the fact that the vessel is big enough to cook the rice, or the vessel has a particular capacity to hold the cooked rice, he will say

स्थाली पचति।

पतञ्जलि in mahAbhA.syam says 'in the absence of a king the senior most minister will enjoy the powers of king.'[4] तद्यथा अमात्यादीनाम् राज्ञा सह समवाये पारतन्त्र्यं व्यवाये स्वातन्त्र्यम्. Now, pada is defined as सुप्तिङन्तम्, so a प्रातिपदिक expressing a कर्त्ता relation in कर्तृवाच्य should have some nominal suffixes else it can not be पद. Therefore, to make such a प्रातिपदिक a पद, प्रथमा विभक्ति is added. पाणिनि has a सूत्र

प्रातिपदिकार्थलिङ्गपरिमाणवचनमात्रे प्रथमा। (2.3.46)

[4]एवं तर्हि प्रधानेन समवाये स्थाली परतन्त्रा, व्यवाये स्वतन्त्रा

Thus in case of **कर्तृवाच्य कर्त्ता** will have **प्रथमा विभक्ति**, and in case of **कर्मवाच्य**, **कर्म** will have a **प्रथमा विभक्ति**. Such a **कारक** which is expressed by the verbal suffix is called **अभिहित**. In case a **कर्त्ता** is not expressed or **अनभिहित**, as in the case of **कर्मवाच्य**, then it gets third case suffix by the **सूत्र** "**कर्तृकरणयोस्तृतीया**".

San: **देवदत्तः पचति।**

gloss: Devadatta cooks.

San: **देवदत्तेन पच्यते।**

gloss: **देवदत्त**{3} cooks. (being cooked by Devadatta).

2.1.2 कर्म

कर्तुरीप्सिततमं कर्म (1.4.49).

What the **कर्त्ता** seeks the most to attain by its action is the karma.

The vibhakti assignment rules are two:

In case the karma is not expressed (anabhihite 2.3.1), **कर्मणि द्वितीया** (2.3.2) assigns second case suffix to the karma.

If the karma is expressed, then karma being abhihita, will get prathamA vibhakti.

San: **देवदत्तः ओदनम् भुङ्क्ते।**

gloss: Devadatta rice{2} eats.

San: **देवदत्तेन पच्यते।**

gloss: devadatta{3} cooks. (being cooked by Devadatta).

2.1.3 करण

साधकतमं करणम् (1.4.42). The factor most effective in the accomplishment of the action is termed as करण or instrument.

कर्तृकरणयोस्तृतीया (2.3.18) assigns third case suffix to the करण.

San: लेखन्या लिखति

gloss: lekhani{3} writes. (writes with pen)

2.1.4 सम्प्रदान

कर्मणा यमभिप्रैति स सम्प्रदानम्। (1.4.32)

That which the agent wants to connect with the object of the action of giving is called recipient. अभिप्रैति means सम्बद्धुमिच्छति – wants to connect. By giving is meant passing of one's ownership of the thing to the recipient. The term सम्प्रदान is meaningful. The two upasargas सम् and प्र give the qualified sense viz. complete transfer.

sampradAna is expressed by the fourth case by the सूत्र "चतुर्थी सम्प्रदाने" (2.3.13)

San: विप्राय गां ददाति।

gloss: brahmin{4} cow gives. (gives cow to a brahmin)

2.1.5 अपादान

The सूत्र "ध्रुवम् अपाये अपादानम्" (1.4.24) states that "when ablation or separation is to be effected by a verbal action, the point of separation is called the ablative". Further the सूत्र "अपादाने पञ्चमी" (2.3.28) states that the ablative case is denoted by the fifth case affix. Thus in the example

San: गोष्टाद् गोवत्सो निर्गच्छति।

gloss: cow-shade{5} calf go_away.

The separation or departure of calf from the cow-shade is marked by the fifth case suffix.

2.1.6 अधिकरण

"आधारः अधिकरणम् (1.4.45)", is the सूत्र describing the locus or अधिकरणṪhe place or thing which is the site or substratum of the action subsisting in the agent or the object and is thereby the immediate site of action is अधिकरणÀccording to भर्तृहरि

कर्तृ-कर्मव्यवहितामसाक्षाद्धारयत्क्रियाम्।

उपकुर्वत् क्रियासिद्धौ शास्त्रे अधिकरणम् स्मृतम्॥

The सूत्र "सप्तम्यधिकरणे च" (2.3.36) assigns seventh case to express the अधिकरण.

San: कटे बालिका अस्ति।
gloss: The girl is on the mat.

Thus in a nutshell, कारक is a semantic relation expressed through the विभक्तिs, and there are 6 such कारक relations.

2.2 उपपद विभक्ति

There are also certain indeclinable words which demand certain case suffixes. Such words are called उपपदs – उपोच्चारितं पदम् – a word pronunced in close proximity. In other words, when a noun takes certain case-affixes by virtue of its being in composition with some other word in the proximity with it, it is called an उपपद विभक्ति.
For example, look at the sentence:
San: रामेण सह सीता वनं गच्छति।
gloss: Rama with Sita forest goes. (Sita goes to forest with Rama).

In this example, राम has third case suffix. This is because of the presence of the word सह. This सह is the word in close proximity (उपपद). This upapada demands that rAma should have third case suffix. Thus this third case suffix does not denote any relation between a noun and a verb.

There are 6 types of उपपद विभक्तिs corresponding to the 6 case suffixes viz. 2, 3, 4, 5, 6 and 7.

The सूत्रs indicating the words demanding these विभक्तिs is given in Appendix II.

Chapter 3

शाब्दबोध

What we have seen so far is how the कारकs are realised in a sentence through the विभक्तिs. The process of understanding the meaning of a sentence, is a reverse process. This process is termed as the process of शाब्दबोधIn India, the three schools viz. व्याकरण, न्याय and मीमांसा differ in the process of शाब्दबोध slightly. The process of शाब्दबोध involves basically the identification of modifier and modified (विशेष्य and विशेषण). According to वैयाकरणs धात्वार्थ (meaning of a verb) is the मुख्य विशेष्य. The नैयायिकs take the प्रथमान्त (the one in first case suffix) as the मुख्य विशेष्य. There is very subtle difference between the मीमांसकs and वैयाकरणs approach. मीमांसकs take भावना as the मुख्य विशेष्य.

We will not go in the theoretical discussions on the differences between these three schools and their consequences, etc. Our main aim is to develop intelligence in the system so that computer can assist human beings in understanding the Sanskrit texts.

If we look at Sanskrit books on various topics, we see that the original संहिता text is followed by simple पदपाठ, followed by the information on various compounds in the text, and finally the anvaya. This अन्वय is done taking into account the सूत्रs which assign different विभक्तिs to कारकs. Look at the example from the 'संक्षेप रामायणम्'(see figure 3.1). Here we see all the steps mentioned above.

श्लोकः

एतदिच्छाम्यहं श्रोतुं परं कौतूहलं हि मे।
महर्षे त्वं समर्थोऽसि ज्ञातुमेवंविधं नरम् ॥ 5 ॥

पदच्छेदः

एतद् इच्छामि अहम् श्रोतुम् परम् कौतूहलम् हि मे।
महर्षे! त्वम् समर्थः असि ज्ञातुम् एवं-विधम् नरम् ॥

पदपरिचयः

पदम्	विश्लेषणम्	पदम्	विश्लेषणम्
एतद्	एतद्- द्० सर्व० नपुं० द्विती० एक०	त्वम्	युष्मद् - सर्व० प्र० एक०
		समर्थः	अ० पुं० प्र० एक०
इच्छामि	इष् - कर्तरि लट्, उपु० एक०	असि	अस् - कर्तरि लट् मपु० एक०
अहम्	अस्मद् - सर्व० प्र० एक०	ज्ञातुम्	तुमुन्नन्तम् अव्ययम्
श्रोतुम्	तुमुन्नन्तम् अव्ययम्	एवम्	एवम् - अव्ययम्
कौतूहलम्	अ० नपुं० प्र० एक०	एवंविधम्	एवंविध - अ० पुं० द्विती० एक०
हि	अव्ययम्, यस्मादित्यर्थे प्रयुक्तम्		समस्तम्
महर्षे !	इ० पुं० सम्बो० प्र० एक०	नरम्	अ० पुं० द्विती० एक०

आकाङ्क्षा

अहम् इच्छामि।

अहं किम् इच्छामि ?	अहं श्रोतुम् इच्छामि।
अहं किं श्रोतुम् इच्छामि ?	अहम् एतद् (पूर्वोक्तगुणसम्पन्नस्य नरस्य विषये) श्रोतुम् इच्छामि।
किमर्थम् इच्छामि ?	हि मे कौतूहलम् (अस्ति)।
कथं (त्वत्तः एव) श्रोतुम् इच्छामि ?	(यतः) त्वम् एवंविधं नरं ज्ञातुं समर्थः असि।

अन्वयः

महर्षे ! अहम् एतत् श्रोतुम् इच्छामि, हि मे परं कौतूहलम् (अस्ति)। त्वं (च) एवंविधं नरं ज्ञातुं समर्थः असि।

Figure 3.1: Sanksheparamayana

The सूत्रs stated in the chapter 2 above are directly relevant and useful for the generation of Sanskrit texts. When a listener listens to a sentence, now what s/he gets is the विभक्तिs. From these विभक्तिs, the listener should analyse and get the कारकs. This task is not an easy task. We give here some examples to illustrate the complexity.

Look at the sentence:

San: रामः वनम् गच्छति।

gloss: Rama forest goes.

The morphological analyser does the analysis of each of these words, and produces the following answers for each of the words.

रामः : राम {1,पुंलिङ्ग, एकवचनम्}

: रा {कर्तरि,लट्,उत्तम पु.,बहुवचनम्,परस्मैपदी}

वनम् : वन {1, नपुं, एकवचनम्}

: वन {2, नपुं, एकवचनम्}

गच्छति : गम् {प्र0. पु0., एक व0; लट्, परस्मै0,}

गच्छति : गम् + शतृ0 + {7, नपुं0, एकवचनम्}

गच्छति : गम् + शतृ0 + {7, पुं0, एकवचनम्}

A human being while reading does not even 'see' these ambiguities. But when we analyse using machine, since machine does not use common sense, world knowledge, etc. shows all possible analysis. So machine has two tasks:

- Deciding the main verb (whether रा or गम्), and
- Assigning the कारक roles.

The question is, whether there is any way to rule out the possibility of रा being a धातु. Following सूत्र by पाणिनि comes to our rescue:
"यस्य च भावेन भावलक्षणम् ।". (2.3.37)
This सूत्र rules out the possibility of the following two analysis of गच्छति.
गच्छति : गम् + शतृ + {7, नपुं0 , एकवचनम्}
गच्छति : गम् + शतृ + {7, पुं0 , एकवचनम्}
This now leaves only one analysis of गच्छति as a verb. Further the verb gam has an expectancy of two कारकs viz. कर्त्ता and karma. Further the word गच्छति is कर्तृवाच्य. Hence कर्त्ता being अभिहित should be in प्रथमा विभक्ति. But there are two पदs viz. both राम and वन in प्रथमा विभक्ति. Hence the question is which one should be taken as कर्त्ता. We postpone this decision, and look at the word with second case suffix to decide the कर्म. We get only one word viz. वनम्. Hence it is assigned a कर्म role. Since any noun can not

have more than one roles. Because, all the सूत्रs describing कारक सज्ञाs are governed by the सूत्र "आकाडारात् एका सज्ञा"(1.4.1). Hence vanam can get only one सज्ञा viz. कर्म. This leaves राम with the कर्त्ता कारक role. Thus we see that various सूत्रs of पाणिनि come into play in assigning the कारक roles to the nouns.

Let us consider the possibility of रा as a verb. The verb रा is in उत्तमपुरुष, and is in कर्तरि प्रयोग. Hence, one may assume the कर्त्ता to be वयम्, and need not be specified. वनम् being in द्वितीया विभक्ति can be कर्म for रा. But then गच्छति will remain without any कारकs. Hence this अन्वय will be ruled out leaving only one अन्वय as follows:

रामः = कर्त्ता of गच्छति ।

वनम् = कर्म of गच्छति ।

We try to follow this process mechanically, taking the help of पाणिनि's सूत्रs. To implement this mechanically, what is needed is a mechanism to solve the constraints simultaneously. We have used the CLIPS for implementing the system. In the fifth chapter we discuss the CLIPS environment followed by the overall architecture.

Chapter 4

Earlier efforts

There have been much enthusiasm in the field of Sanskrit computations since the advent of computers. There have been efforts to demonstrate suitability of Sanskrit shastra for automatic language processing as well as to develop computational tools for processing Sanskrit texts. The few notable efforts of the first type are by the Akshar Bharati group(Bharati,1994). The latter have been tried by various individuals and groups – Huet(2008), Scharf(2008), Jha(2008), and Goel(2007), apart from the department of Sanskrit Studies in collaboration with Rashtriya Sanskrit Vidyapeetha, Tirupati, to name a few. There are also ongoing efforts to model the paninian process of analysis through computers by Sharf(2008), Mishra(2009), Goel(2008), and the package गणक - अष्टाध्यायी (स्वामि तरल्बालु,2009). The majority of these han-

dle morphology reasonably well. Only few of them have been handling the parsing at sentence level.

Parsers based on Paninian Grammar was developed by Bharati (1995). This parser was developed following the Integer programming approach, and later on improved by adapting the bipartite graph model. This Parser was developed for Hindi and not for Sanskrit. The parser required kAraka charts for each verb describing the various कारक roles that are mandatory for the verb (i.e. the expectancy), and the corresponding विभक्तिs the corresponding noun takes.

A कारक chart essentially marks the necessity of a कारक (whether it is mandatory, desirable or optional), and shows the corresponding vibhakti. The mandatory कारकs are कर्त्ता and karma in case of transitive verbs, and कर्त्ता in case of intransitive verbs. सम्प्रदान is also mandatory in case of certain verbs, whose list is available through the पाणिनि's सूत्रs. Further the desirable कारकs are the करण andअपादान. Most of the verbs have an expectancy for these कारकs. अधिकरण being the location or आधार for either कर्त्ता or कर्म, is a कारक which any verb can have. At the same time, unless necessary, one does not mention it. Hence it is optional.

Now, if we look at the पाणिनि's way of organisation, we see that, Panini assumes the knowledge of sakarmana/akarmaka while framing his rules. We

refer to XawuprakASaH (in press) for the कर्माकाङ्क्षा information. Next, the list of verbs desiring some कारकs such as सम्प्रदान, अपादान etc. is also listed separately. So with this information, we decided to use the पाणिनि's सूत्रs only and develop a rule based parser, instead of using the कारक charts.

Chapter 5

About CLIPS

5.1 Introduction

Making computer think like people is Artificial Intelligence. In our effort to develop a parser, we will be training a computer to 'think' like a human being and decide the कारक relations. An expert system or an intelligent system is a special system that uses Artificial intelligence techniques. A typical expert system consists of

- Knowledge Base (Rules)

  The knowledge base is in the form of IF THEN rules. For example,

IF

the noun has सम्प्रदान role

THEN

it will take चतुर्थी विभक्ति.

Here the section of the rule between IF and THEN viz. 'the noun has सम्प्रदान role' is called an antecedent or conditional part or pattern part or Left Hand Side (LHS). The part after THEN is called the consequence, or the Right Hand Side (RHS). When this condition is satisfied, or the pattern is matched with any of the existing facts from the memory, then rule is said to be fired.

- Facts

  The facts is the actual working memory. It varies with the inputs, and the state of the system.

- Inference Engine

  The inference engine makes inferences by deciding which rules are satisfied by the facts. All the rules for which the facts match the LHS, are said to be activated or satisfied. All these rules will be on the agenda. In such cases, the inference engine should choose one of the rules. This selected rule then is fired. The selection of the rules is called conflict

resolution. The conflict resolution may be either explicit by prioritizing the rules, say by assigning the saliency, or implicit based on the specificity of conditions etc. In case the rules are provided with a saliency, a rule with highest priority is selected by the inference engine.

The expert system language that I have used in my thesis is called CLIPS which is an acronym for C Language Integrated Production System. CLIPS was designed at NASA/Johnson Space Center with the specific purposes of providing high portability, low cost, and easy integration with external system. CLIPS was written using the C programming language to facilitate these objectives.

The basic elements of CLIPS, as of any expert system, are

- Fact-list: global memory for data.
- knowledge-base:contains all the rules.
- Inference engine:controls overall execution.

5.1.1 Facts

Each chunk of information is called a Fact. Facts consist of:
a name with zero or more fields for associated values. For example, the following fact defines a name or a field called 'sentence', whose value is 'रामः

ग्रामम् गच्छति ।'

(sentence (rAmaH grAmam gacCati))

This is a simple fact. Consider another example: The morphological analysis of the word rAme.na consists of several facts viz. rAma is the prAtipadika, its gender is masculine, the word is in the singular number, and it is in third case ending. This is represented in CLIPS as follows

(word rAmeNa)

(number singular)

(vibhakti third)

(gender masculine)

The problem with such kind of representation is, if we have several words, then whose number is what, which one is singular, which one is masculine is not recorded. The association of the word with its features is gone. To avoid it, we make use of structured facts. To use the structured facts, first CLIPS should be informed about the new structure. This is done through the DEFTEMPLATE.

5.1.2 Deftemplate

Before facts can be constructed, CLIPS must be informed of the list of valid slots for a given relation name. A deftemplate is used to describe groups of facts sharing the same relation name and contain common information. The deftemplate for subanta is given below.

```
(deftemplate sup
       (slot id)
       (slot word)
       (slot rt)
       (slot lingam)
       (slot viBaktiH)
       (slot vacanam)
       (slot kAraka))
```

The morpological analysis for the word, say rAmebhyaḥ, now will be a fact. This analysis will have the template structure of sup. The fact is then declared or stored in the working memory as
(deffacts (sup
(id 1) (word rAmeByaH) (rt rAma) (lingam puM) (viBakwiH 4) (vacanam bahu)))

Facts with a relation name defined using deftemplate are called deftemplate facts.

5.2 Basics of CLIPS operations

In this section we give the basic commands to handle CLIPS.

- Entering and Exiting CLIPS

  1. Once the CLIPS has been installed, the command 'clips', should activate the the CLIPS shell.
  2. The CLIPS prompt is: 'CLIPS>' where command can be entered directly to CLIPS is called the top level mode.
  3. To exit CLIPS 'or' The normal mode of leaving CLIPS is with the exit command. The syntax of this command is (exit)

- Displaying Facts:

  The facts command can be used to display the facts in the fact-list. The syntax of the fact command is 'CLIPS> (facts)'.

- Adding and Removing Commands: All facts known to CLIPS are stored in the fact-list. To add a new fact to the fact-list, we can use

the assert command. The syntax is assert command: 'CLIPS> (assert)'

Example: (assert (viBakwiH 1) (vacanam eka))

Just as facts can be added to the fact-list, they can also be removed. Removing facts from the fact-list is called retraction and is done with the retract command. The syntax for retract is:
'CLIPS> (retract <integer>)'
'(clear)' or '(retract *)' will remove all the facts.

- Modifying Facts:
  Slot values of deftemplate facts can be modified using the modify command.

  Example: (modify ?s1 (kAraka karwA))

- Rules: Rules can be typed directly into CLIPS or they can be loaded in from a file of rules created by an editor. In CLIPS, the 'IF cond THEN action' is written as
  cond

=>

action

Thus, explicitly the words IF and THEN are not written. The part before the => is the condition and the part after it is the action.

- Execution and Agenda:

  After specifying the rules and providing the action, to execute the rules i.e. to run the CLIPS program, use the run command.

  The syntax of the run command is: CLIPS¿ (run)

  When the program runs, and there is a conflict, then the rule with the highest salience on the agenda is fired.

  All the rules that satisfy the conditions, are on agenda. To display the rules on the agenda, use agenda command: 'CLIPS¿ (agenda)'

  An agenda command would produce the following output.

- Reset:

  The reset command is the key method for starting or restarting. Facts asserted by a reset satisfy the patterns of one or more rules and place activation of these rules on the agenda.

- The printout command is used to print information.

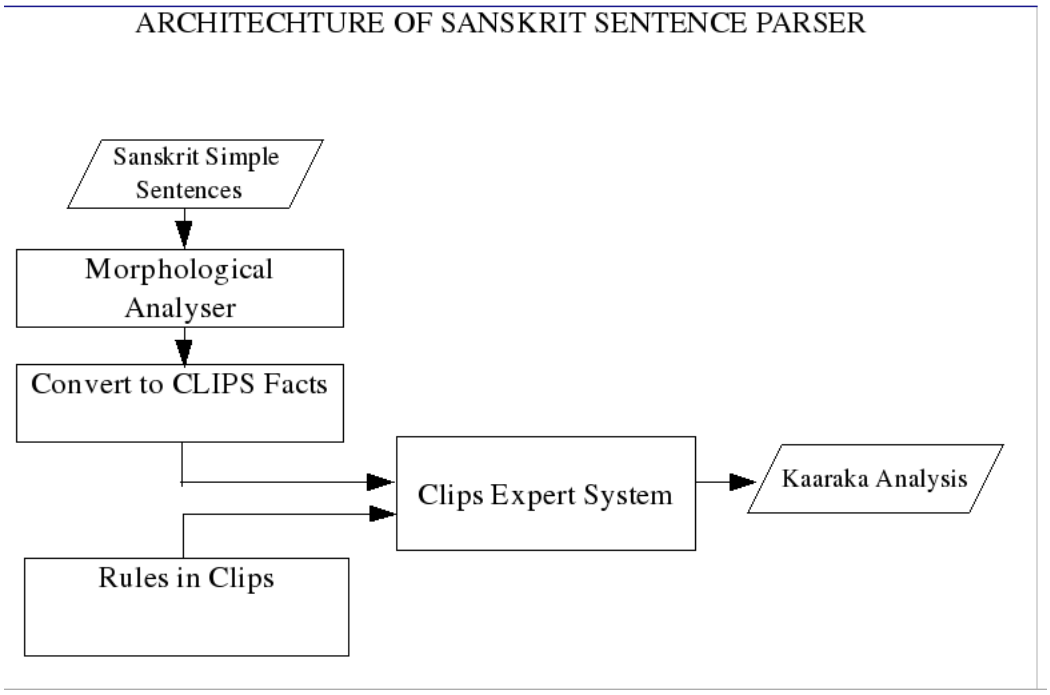
Example: CLIPS¿ (printout t ‘stop’ crlf)

‘t’ stands for the standard output device.

- ‘load’ command is used for loading of rules from an external file.
- ‘save’ command is opposite of load, and allows saving of constructs to disk.
- save-fact will save all facts in the fact-list in file and load-fact will load it.

5.3 Architecture

The flow diagram of the system is given in figure 5.5. The input for our parser is a Sanskrit text with single finite verb (ti”n), and the output is its कारक analysis. The main purpose of this exercise is to do only कारक and उपपदविभक्ति analysis. The input sentence is passed through the morphological analyser to get the word level analysis. This analysis is then converted into CLIPS facts. The facts, which vary with a sentence, along with the rules are then passed to the CLIPS interpreter for the कारक analysis.



5.4 Sample input

रामः वनम् गच्छति।

5.5 Sample output

रामः = कर्त्ता of गच्छति

वनम् = कर्म of गच्छति

Chapter 6

Facts and Rules

In this chapter we describe the templates used for declaring the facts, followed by the rules. The structure of templates is largely influenced by the morphological analyser output. पाणिनि classifies the words into two types: सुबन्त and तिङन्त. But this classification is not sufficient for our purpose. Consider the क्त्वान्त word say 'गत्वा'. If we just mark it as a सुबन्त, or to be more specific, say an अव्यय, it is not sufficient. The reason being, the underlying verb 'गम्', has its own expectancies. And to know that it has expectancies, the machine should know that the there is an underlying verb, and it has certain expectancies. Hence we define the templates for सुबन्त, तिङन्त, अव्यय, तद्धित and कृदन्तs. We give below the slots available in each of them. These slots correspond to the features associated with that form.

We also make a provision for the slot corresponding to the कारक analysis, which will be filled in by the CLIPS inference engine. In case of verbs, slotes for the corresponding कारक positions are provided.

6.1 Fact Templates

```
(deftemplate sup
(slot id) ; To store the word number
(slot mid); To store the analysis id
            (since in principle, there can be more than one ids)
(slot word); To store the given word
(slot rt) ; to store its pratipadika
(slot lingam) ; to store the linga
(slot vibhaktiH) ; to store the vibhakti
(slot vacanam) ; to store the vacana
(slot kAraka)) ; to store the kAraka role

(deftemplate tin
(slot id)
(slot mid)
(slot word)
```

```
(slot rt)

(slot dhatuH)

(slot lakAraH)

(slot prayogaH)

(slot purushaH)

(slot vacanam)

(slot padI)

(slot gaNaH)

(slot karttA_pos)

(multislot karma_pos)

(slot karaNa_pos)

(slot sampradAna_pos)

(slot apAdAna_pos)

(slot adhikaraNa_pos))


(deftemplate krt ; For kridantas

(slot id)

(slot mid)

(slot word)

(slot krt_pratyayaH)

(slot lingam)
```

```
(slot vibhaktiH)

(slot vacanam)

(slot rt)

(slot dhatuH)

(slot gaNaH)

(slot karttA_pos)

(slot karma_pos))


(deftemplate avy

(slot id)

(slot mid)

(slot word))


(deftemplate taddhita

(slot id)

(slot mid)

(slot word)

(slot rt)

(slot lingam)

(slot vibhaktiH)

(slot vacanam))
```

6.2 Rules

We process the input in two steps. In the first step, wherever possible, we try to remove irrelevant morphological analysis. This then helps us in reducing the search space, and also if machine fails, then it helps the human being to rule out some of the possibilities. For example, as explained earlier, in the sentence 'रामः vanam गच्छति', the analysis of the word गच्छति as a subanta is irrelevant. Hence we remove or retract it. Similarly, there are many cases, where a word is ambiguous between the second case or first case and also the सम्बोधन. The सम्बोधन will be irrelevant only if certain special conditions are met. For example, consider the sentence

बालः खादति ।

In the sentence बालः खादति । the word खादति can have the following analysis खादति स्त्री 8 एक0 where खादति is the feminine शतृ form. But in this context, since their is no verb with लोट् लकार or विधिलिङ् लकार (see appendix **Assign Karaka Rules** for more details), this analysis is ruled out.

We illustrate below one rule corresponding to the सूत्र

"यस्य च भावेन भावलक्षणम्" (2.3.37).

Second step is to actually assign कारक roles. Now the विभक्तिs may be either

upapada or कारक. We first mark the words with उपपद विभक्तिs, since these are just next to the given words. After this, then we assign the कारक विभक्तिs. Assigning कारक विभक्तिs is not an easy task. Because several times a word has more than one morphological analysis possible. In such cases, we have to look at the assignments simultaneously, and not sequentially. as explained earlier.

We describe below three rules, one for removing the irrelevant morphological analysis, the second one for marking the upapada विभक्तिs and the third one for marking the कारक विभक्तिs.

6.3 Rule Description

6.3.1 Rule for retracting an analysis

In this section we describe a rule that retracts the seventh case analysis when it is irrelevant.

- Motivation for the necessity of a rule

  Look at the sentence:

  rAmaḥ vanam गच्छति.

  Here in this sentence गच्छति has three possible analysis, viz -

गच्छति : गम् {प्र0. पु0., एक व., लट्, परस्मै0,}

गच्छति : गम् + शतृ + {7, नपुं0, एकवचनम्}

गच्छति : गम् + शतृ + {7, पुं0, एकवचनम्}

The second and the third analysis are not relevant in this context. Hence these analysis may be ruled out on the basis of पाणिनि's सूत्र "यस्य च भावेन भावलक्षणम्" (2.2.37). The meaning of this सूत्र is:

If an activity aims at another activity then the verb denoting the first activity will have seventh case (naturally after a कृदन्त). In such cases either the कर्त्ता or the कर्म or both of this first activity will also be then in seventh case.

- Rule in CLIPS

```
(defrule sati-saptami
(test (>= (count-sati-saptami krt) 1))
;if there is at least one {\dn E\387wd\306wt} with 7 {\dn EvBE\3C4w}
=>
(do-for-all-facts
((?w kqw))
;for all k.rdantas
(eq ?w:krt_pratyayaH satri)
```

```
;with satri pratyaya
(= ?w:vibhaktiH 7)
;and 7 vibhakti analysis
(not (any-factp ((?s sup)) (= ?s:vibhaktiH 7)))
;if there is no other subanta with 7 vibhakti
(any-factp ((?w1 sup tin)) (= ?w1:id ?w:id)))
;and the krdanta has at least one more analysis
other than satri + 7 vibhakti
(retract ?w)
;remove the satri + 7 analysis
(printout foo "(" ?w:id " " ?w:mid ")
  yasya ca bhAvena bhAvalakshaNam")crlf )
)
)
```

- Explanation

  Thus we observe that, in the sentence रामः वनं गच्छति', there is a कृदन्त (viz.गच्छति) in seventh विभक्ति. Therefore the condition of the rule is fulfilled and hence the rule is fired. The action part checks every कृदन्त with शतृ+7 विभक्ति analysis. In the given sentence there is only one k.rdanta गच्छति satisfying this condition. Further, the next condition that there be no other word with seventh विभक्ति is also satisfied and

finally this word has one more analysis viz as a तिङन्त. Hence the शतृ+7 analysis of the word गच्छति is deleted.

- Input

  Only the relevant part of the input is produced here.

  (deffacts morph_information

  (tin (id 1) (mid 1) (word rAmaH) (rt rA1)(prayogaH karwari)(lakAraH la.t)(puruRaH u)(vacanam bahu)(paxI parasmEpaxI)(XAwuH rA)(gaNaH axAxiH))

  (sup (id 1) (mid 2) (word rAmaH) (rt rAma)(lifgam puM)(viBakwiH 1)(vacanam eka))

  (sup (id 2) (mid 1) (word vanam) (rt vana)(lifgam napuM)(viBakwiH 1)(vacanam eka))

  (sup (id 2) (mid 2) (word vanam) (rt vana)(lifgam napuM)(viBakwiH 2)(vacanam eka))

  (krt (id 3) (mid 1) (word gacCawi) (rt gacCaw)(lifgam puM)(viBakwiH 7)(vacanam eka) (kqw_prawyayaH Sawq)(XAwuH gamLz)(gaNaH BvAxiH))

  (krt (id 3) (mid 2) (word gacCawi) (rt gacCaw)(lifgam napuM)(viBakwiH 7)(vacanam eka) (kqw_prawyayaH Sawq)(XAwuH gamLz)(gaNaH BvAxiH))

(krt (id 3) (mid 3) (word gacCawi) (rt gacCawI)(lifgam swrI)(viBakwiH 8)(vacanam eka))

(tin (id 3) (mid 4) (word gacCawi) (rt gam1)(prayogaH karwari)(lakAraH la.t)(puruRaH pra)(vacanam eka)(paxI parasmEpaxI)(XAwuH gamLz)(gaNaH BvAxiH))

)

- Execution

  When we execute the rules with the given facts, and check the new facts, we get the following

```
(agenda)
0      sati-saptami: f-0
For a total of 1 activation.
CLIPS> (run)
CLIPS> (facts)
```

The facts are:

f-0 (initial-fact) f-1 (wif (id 1) (mid 1) (word rAmaH) (rt rA1) (XAwuH rA) (lakAraH la.t) (prayogaH karwari) (puruRaH u) (vacanam bahu) (paxI parasmEpaxI) (gaNaH axAxiH) (karwA_pos nil) (karma_pos nil) (karaNa_pos nil) (sampraxAna_pos nil) (apAxAna_pos nil) (aXikaraNa_pos

nil))

f-2 (sup (id 1) (mid 2) (word rAmaH) (rt rAma) (lifgam puM) (viBakwiH 1) (vacanam eka) (कारक nil) (viSeRaNa_pos nil))

f-3 (sup (id 2) (mid 1) (word vanam) (rt vana) (lifgam napuM) (viBakwiH 1) (vacanam eka) (कारक nil) (viSeRaNa_pos nil))

f-4 (sup (id 2) (mid 2) (word vanam) (rt vana) (lifgam napuM) (viBakwiH 2) (vacanam eka) (कारक nil) (viSeRaNa_pos nil))

f-7 (kqw (id 3) (mid 3) (word gacCawi) (kqw_prawyayaH nil) (lifgam swrI) (viBakwiH 8) (vacanam eka) (rt gacCawI) (XAwuH nil) (gaNaH nil) (karwA_pos nil) (karma_pos nil))

f-8 (wif (id 3) (mid 4) (word gacCawi) (rt gam1) (XAwuH gamLz) (lakAraH la.t) (prayogaH karwari) (puruRaH pra) (vacanam eka) (paxI parasmEpaxI) (gaNaH BvAxiH) (karwA_pos nil) (karma_pos nil) (karaNa_pos nil) (sampraxAna_pos nil) (apAxAna_pos nil) (aXikaraNa_pos nil))

For a total of 7 facts.

Thus we see that rule has deleted the following two facts corresponding to the kqxanwas.

(kqw (id 3) (mid 1) (word gacCawi) (rt gacCaw)(lifgam puM)(viBakwiH 7)(vacanam eka) (kqw_prawyayaH Sawq)(XAwuH gamLz)(gaNaH BvAxiH))

(kqw (id 3) (mid 2) (word gacCawi) (rt gacCaw)(lifgam napuM)(viBakwiH 7)(vacanam eka) (kqw_prawyayaH Sawq)(XAwuH gamLz)(gaNaH BvAxiH))

Similarly I have written rules for retracting the सम्बोधन analysis whenever it is irrelevant. The actual coding is available in Appendix "(assignkAraka)"

6.3.2 Rule for marking upapada vibhakti

There are six rules corresponding to six upapada विभक्तिs. We look at a very frequent case of 'saha'. The पाणिनि's सूत्र is "सहयुक्तेऽप्रधाने" (2.3.19). The implementation of this rule is given below.

- Motivation for the necessity of a rule

  Look at the sentence:

  रामेण सह सीता वनं गच्छति।

  Here in this sentence सह assigns the third case suffix to राम. We write the rule as follows.

- Rule in CLIPS

```
(defrule assign_trtiyA-upapada
```

```
; Define the rule
(declare salience 100)
; priority of this rule is 100
(test (> (count-trtIyAnta sup) 0))
; check if third case vibhakti word exists
(test (> (count-tqtIyA-upapada avy) 0))
; check if there is an indeclinatable demanding the third case.
=> ; execute the following
(delayed-do-for-all-facts
; for all pair of facts such that
((?a avy) (?s sup))
; one is a subanta and the other an avyaya
(and (= ?s:vibhaktiH 3) (= (- ?a:id ?s:id) 1)
(eq (gdbm_lookup "tqtIyA_upapada_list.gdbm" ?a:word) "1"))
; If the subanta has thritiyA vibhakti, the avyaya is adjacent to it and the a
(modify ?s ({\dn kArk} upapala_vibhaktiH))
; then mark the noun as upapada vibhakti
(printout bar "(" ?s:id " " ?a:word " kI upapaxa_viBakwiH)" crlf)))
; print it to a file for further processing
```

- Explanation

It is self explanatory. We declare the salience of this rule to be higher, so that when there is a conflict, this rule gets priority over the other rules.

6.3.3 Rule for marking कारक roles

Here we describe a rule for marking the कर्त्ता in case of karmavAchya verb forms. In case of कर्मवाच्य, both the कर्त्ता as well as करण take the same विभक्ति viz. तृतीया. As in 'रामेण बाणेन वालिः हन्यते'. It is difficult for a machine to decide which is करण and which is कर्त्ता, just by looking at the words or their analysis. What is needed is extra-linguistic information that बाण is an instrument. Since our system does not have such a knowledge yet, we mark both the words as 'कर्त्ता-करण-वा'.

We give below the rule that marks the abhihita as a karma, and the words in third case as 'कर्त्ता-kara.na'. The rule in CLIPS is with self explanatory comments.

```
(defrule assign_{\dn k\381wA\0}-karaNa-vA_karma_karmavAcy
(test (> (count-prathamAnta sup) 0))
; at least one word in nominative case exists
(test (> (count-tqtIyAnta sup) 0))
; at least one word in third case
```

```
(test (eq (karmaNi-vA tin) TRUE))
; at least one tin in karmani
=>; execute the following actions
(delayed-do-for-all-facts; for each of the triplate
((?s1 sup) (?s2 sup) (?w tin))
; with two subantas and one tinganta
(and (= ?s1:viBakwiH 3) (= ?s2:viBakwiH 1)
(<> ?s1:id  ?s2:id  ?w:id)
(eq ?w:prayogaH karmaNi))
; such that one sup in third case, one in nominative
; and the tin in karmani prayoga
(modify ?w ({\dn k\381wA\0}_vA_karaNa_pos ?s1:id)
(karma_pos ?s2:id))
;mark the karta and karma positions
(modify ?s2 ({\dn kArk} karma))
; mark one in nominative as karma
(modify ?s1 ({\dn kArk} {\dn k\381wA\0}_vA_karaNa))
; mark the one in third case as {\dn k\381wA\0}_vA_karaNa
; print the same to a file for further processing
(printout bar "(" ?s1:id " "  ?w:word " kA {\dn k\381wA\0}_vA_karaNa)" crlf)
(printout bar "(" ?s2:id " " ?w:word " kA karma)" crlf)
```

```
))
```

Chapter 7

Conclusion

The purpose of my thesis is to build a parser for Sanskrit. To decide the suitability of CLIPS shell for developing a parser, as a pilot study, I collected various kinds of sentences with single तिङन्त. These sentences had the examples from both the उपपद विभक्तिs as well as कारक विभक्तिs. I used the in-house morphological analyser for my work. I also decided not to take the णिजन्त as the morphological analyser was not yet fully functional for the same. Further I also did not consider the षष्ठी विभक्ति, since it is used to denote a कारक as well as noun-noun relations.

I implemented approximately 15 rules for the analysis, and could run all types of sentences satisfactorily. The appendix lists all the sentences, that were tested on the machine.

To have a full-fledged realistic parser for Sanskrit, still there is a long way to go. With the current experience of building a simple parser, I think it is feasible to build a full-fledged parser for Sanskrit.

द्वितीया-उपपद-तालिका:=>

प्रति	उभयतः	सर्वतः	धिक्	उपर्युपरि	अध्यधि	अधो-अधः	अधि-अधि
परितः	समया	निकषा	अन्तरा	अन्तरेण	अनु	यावत्	ऋते
उत्तरेण	हा	दक्षिणेन	विना	पृथक्	नाना		

"उभसर्वतसो कार्या धिगुपर्यादिषु त्रिषु।
द्वितीया आम्रेडितान्तेषु ततो अन्यत्रापि दृश्यते" ॥ (वा० २/३/२)
"अभितःपरितःसमयानिकषा-हा-प्रतियोगेषु च दृश्यते" (वा० २/३/२)
"अन्तरान्तरेणयुक्ते" (पा०सू० २/३/४)
"एनपा द्वितीया" (पा०सू० २/३/३१)
"पृथग्विनानानाभिस्तृतीयान्यतरस्याम्" (पा०सू० २/३/३२)
"अनुर्लक्षणे" (पा०सू० १/४/८४)
"कर्मप्रवचनीययुक्ते द्वितीया" (पा०सू० २/३/८)
"लक्षणेत्थम्भूताख्यानभागवीप्सासु प्रतिपर्यनवः" (पा०सू० १/४/८९)
"अभिरभागे" (पा०सू० १/४/९०)
"दूरान्तिकार्थैभ्यो द्वितीया च" (पा०सू० २/३/३५)
"कालाध्वनोरत्यन्तसंयोगे" (पा०सू० २/३/५)

तृतीया-उपपद-तालिका:=>

सह	साकम्	समम्	सार्धम्	अलम्

"सहयुक्ते अप्रदाने" (पा०सू० २/३/१९)
"अपवर्गे तृतीया" (पा०सू० २/३/६)

चतुर्थी-उपपद-तालिका: =>

नमः स्वस्ति स्वाहा स्वधा अलम् वषट् हित क्लृपि सम्-पद भू

"नमःस्वस्तिस्वाहास्वधालंवषड्योगाच्च" (पा०सू० २/३/१६)
"मन्यकर्मण्यनादरे विभाषा अप्राणिषु (
"तादर्थ्य उपसंख्यानम्"(वा० २/३/१३)
"क्लृपि सम्पद्यमाने चतुर्थी वक्तव्या" (वा० २/३/१३)
"हितयोगे चतुर्थी वक्तव्या" (वा० २/३/१३)
"तुमर्थाच्च भाववचनात्" (पा०सू० २/३/१५)

पञ्चमी-उपपद-तालिका:=>

अन्य आरात् इतर ऋते पूर्व उत्तर दक्षिण प्राक् प्रत्यक् दक्षिणा उत्तरा दक्षिणाहि उत्तराहि पृथक् विना नाना
स्तोक अल्प कृच्छ्र कतिपय दूर विप्रकृष्ट अन्तिक समीप अभ्याश आरभ्य बहिः प्रभृति भिन्न

"अन्यारादितरर्तेदिक्छब्दाञ्चूत्तरपदाजाहियुक्ते" (पा०सू० २/३/२९)
"पृथग्विनानानाबिस्तृतीयान्यतरस्याम्" (पा०सू० २/३/३२)
"अकर्तर्यृणे पञ्चमी" (पा०सू० २/३/२४)
"करणे च स्तोकाल्पकृच्छ्रकतिपयस्यासत्ववचनस्य" (पा०सू० २/३/३३)
"दूरान्तिकार्थैः षष्ठ्यन्यतरस्याम्" (पा०सू० २/३/३४)
"दूरान्तिकार्थेभ्यो द्वितीया च" (पा०सू० २/३/३५)

षष्ठी-उपपट्-तालिका:=>

दक्षिणतः उत्तरतः उत्तरेण दक्षिणेन उपरिष्टात् उपरि पुरः पुरस्तात् अन्तः अधः अधस्तात् स्वामिन् ईश्वर अधिपति दायाद साक्षिन् प्रसूत प्रतिभू

"षष्ठी शेषे"(पा०सू०२/३/५०)
"स्वामीश्वराधिपतिदायादसाक्षिप्रतिभूप्रसूतैश्च"(पा०सू०२/३/३९)
"आयुक्तकुशलाभ्यां चासेवायाम्"(पा०सू०२/३/४०)

सप्तमी-उपपट्-तालिका:=>

स्वामिन् ईश्वर अधिपति दायाद साक्षिन् प्रतिभू प्रसूत आयुक्त कुशल साधु असाधु निपुण प्रसित उत्सुक नक्षत्र पूजा

"स्वामीश्वराधिपतिदायादसाक्षिप्रतिभूप्रसूतैश्च"(पा०सू०२/३/३९)
"आयुक्तकुशलाभ्यां चासेवायाम्"(पा०सू०२/३/४०)
"साध्वनिपुणाभ्यामर्चायां सप्तम्यप्रतेः"(पा०सू०२/३/४३)
"प्रसुतोत्सुकाभ्यां तृतीया च"(पा०सू०२/३/४४)
"नक्षत्रे च लुपि"(पा०सू०२/३/४५)

विभक्तिसूत्राणि:=>

"अनभिहिते।"(पा० सू० २/३/१)
"कर्मणि द्वितीया।"(पा० सू० २/३/२)
"तृतीया च होश्छन्दसि।"(पा० सू० २/३/३)
"अन्तरान्तरेणयुक्ते।"(पा० सू० २/३/४)
"कालाध्वनोरत्यन्तसंयोगे।"(पा० सू० २/३/५)
"अपवर्गे तृतीया।"(पा० सू० २/३/६)
"सप्तमीपञ्चम्यौ कारकमध्ये।"(पा० सू० २/३/७)
"कर्मप्रवचनीययुक्ते द्वितीया।"(पा० सू० २/३/८)
"यस्मादधिकं यस्य चेश्वरवचनम् तत्र सप्तमी।"(पा० सू० २/३/९)
"पञ्चम्यपाङ्परिभिः।"(पा० सू० २/३/१०)
"प्रतिनिधिप्रतिदाने च यस्मात्।"(पा० सू० २/३/११)
"गत्यर्थकर्मणि द्वितीयाचतुर्थ्यौ चेष्टायामनध्वनि।"(पा० सू० २/३/१२)
"चतुर्थी सम्प्रदाने।"(पा० सू० २/३/१३)
"क्रियार्थोपपदस्य च कर्मणि स्थानिनः।"(पा० सू० २/३/१४)
"तुमर्थाच्च भाववचनात्।"(पा० सू० २/३/१५)
"नमःस्वस्तिस्वाहास्वधाऽलं वषड्योगाच्च।"(पा० सू० २/३/१६)
"मन्यकर्मण्यनादरे विभाषाऽप्राणिषु।"(पा० सू० २/३/१७)

"कर्तृकरणयोस्तृतीया।"(पा० सू० २/३/१८)
"सहयुक्तेऽप्रधाने।"(पा० सू० २/३/१९)
"येनाङ्गविकारः।"(पा० सू० २/३/२०)
"इत्थम्भूतलक्षणे।"(पा० सू० २/३/२१)
"संज्ञोऽन्यतरस्यां कर्मणि।"(पा० सू० २/३/२२)
"हेतौ।"(पा० सू० २/३/२३)
"अकर्त्तर्यृणे पञ्चमी।"(पा० सू० २/३/२४)
"विभाषा गुणेऽस्त्रियाम्।"(पा० सू० २/३/२५)
"षष्ठी हेतुप्रयोगे।"(पा० सू० २/३/२६)
"सर्वनाम्नस्तृतीया च।"(पा० सू० २/३/२७)
"अपादाने पञ्चमी।"(पा० सू० २/३/२८)
"अन्यारादितरर्तेदिक्छब्दाञ्चूत्तरपदाजाहियुक्ते।"(पा० सू० २/३/२९)
"षष्ठ्यतसर्थप्रत्ययेन।"(पा० सू० २/३/३०)
"एनपा द्वितीया।"(पा० सू० २/३/३१)
"पृथग्विनानानाभिस्तृतीयाऽन्यतरस्याम्।"(पा० सू० २/३/३२)
"करणे च स्तोकाल्पकृच्छ्रकतिपयस्यासत्ववचनस्य।"(पा० सू० २/३/३३)
"दूरान्तिकार्थैः षष्ठ्यन्यतरस्याम्।"(पा० सू० २/३/३४)
"दूरान्तिकार्थेभ्यो द्वितीया च।"(पा० सू० २/३/३५)

"सप्तम्यधिकरणे च।"(पा० सू० २/३/३६)
"यस्य च भावेन भावलक्षणम्।"(पा० सू० २/३/३७)
"षष्ठी चानादरे।"(पा० सू० २/३/३८)
"स्वामीश्वराधिपतिदायादसाक्षिप्रतिभूप्रसूतैश्च।"(पा० सू० २/३/३९)
"आयुक्तकुशलाभ्यां चासेवायाम्।"(पा० सू० २/३/४०)
"यतश्च निर्द्धारणम्।"(पा० सू० २/३/४१)
"पञ्चमी विभक्ते।"(पा० सू० २/३/४२)
"साधुनिपुणाभ्यामर्चायां सप्तम्यप्रतेः।"(पा० सू० २/३/४३)
"प्रसितोत्सुकाभ्यां तृतीया च।"(पा० सू० २/३/४४)
"नक्षत्रे च लुपि।"(पा० सू० २/३/४५)
"प्रातिपदिकार्थलिङ्गपरिमाणवचनमात्रे प्रथमा।"(पा० सू० २/३/४६)
"सम्बोधने च।"(पा० सू० २/३/४७)
"साऽऽमन्त्रितम्।"(पा० सू० २/३/४८)
"एकवचनम् सम्बुद्धिः।"(पा० सू० २/३/४९)
"षष्ठी शेषे।"(पा० सू० २/३/५०)

अपादान-कारक-सूत्राणि:=>

कारके(पा०सू० १/४/२३)
ध्रुवमपाये अपादानम्(पा०सू० १/४/२४)
"भीत्रार्थानां भयहेतुः"(पा०सू० १/४/२५)
"पराजेरसोढः"(पा०सू० १/४/२६)
"वारणार्थानामीप्सितः"(पा०सू० १/४/२७)
"अन्तर्द्धौ येनादर्शनमिच्छति"(पा०सू० १/४/२८)
"आख्यातोपयोगे"(पा०सू० १/४/२९)
"जनिकर्तुः प्रकृतिः"(फा०सू० १/४/३०)
"भुवः प्रभवः"(पा०सू० १/४/३१)

सम्प्रदान-कारक-सूत्राणि:=>

"कर्मणा यमभिप्रैति स सम्प्रदानम्"(पा०सू० १/४/३२)
“रुच्यर्थानाम प्रीयमाणः"(पा०सू० १/४/३३)
“श्लाघह्नुङ्स्थाशपां ज्ञीप्स्यमानः"(पा०सू० १/४/३४)
“धारेरुत्तमर्णः"(पा०सू० १/४/३५)
“स्पृहेरीप्सितः"(पा०सू० १/४/३६)
“क्रुधद्रुहेर्ष्यासूयार्थानां यं प्रति कोपः"(पा०सू० १/४/३७)
"क्रुधद्रुहोरुपसृष्टयोः"(पा०सू० १/४/३८)
“राधीक्ष्योर्यस्य विप्रश्नः"(पा०सू० १/४/३९)
“प्रत्यङ्भ्यां श्रुवः पूर्वस्य कर्त्ता"(पा०सू० १/४/४०)
“अनुप्रतिगृणश्च"(पा०सू० १/४/४१)

करण-कारक-सूत्राणि:=>

“साधकतमं करणम्"(पा०सू० १/४/४२)
“दिवः कर्म च"(पा०सू० १/४/४३)

अधिकरण-कारक-सूत्राणि:=>

"आधरः अधिकरणम्"(पा०सू० १/४/४५)
“अधिशीङ्स्थासां कर्म"(पा०सू० १/४/४६)

कर्म-कारक-सूत्राणि:=>

“कर्तुरीप्सिततमं कर्म"(पा०सू० १/४/४९)
“तथायुक्तं चानिप्सतम्"(पा०सू० १/४/५०)
“अकथतं च"(पा०सू० १/४/४१)

कर्ता-कारकम्:=>

“स्वतन्त्रः कर्त्ता"(पा०सू० १/४/५४)

Assign Karaka Rules

```
(reset)
;(facts)
; Get the number of sup entries with praWamA
(deffunction count-praWamAnwa (?template)
(length (find-all-facts ((?fct ?template))
(= (fact-slot-value ?fct viBakwiH) 1))))
;check karwari
(deffunction karwari-vA (?template)
(any-factp ((?fct ?template))
(eq (fact-slot-value ?fct prayogaH) karwari)))
;Get the number of verbs with xvikarmaka AkAfkRA
(deffunction count-xvikarmaka-AkAfkRA (?template)
(length (find-all-facts ((?fct ?template))
(eq (gdbm_lookup "xvikarmaka_XAwu_list.gdbm" (fact-slot-value ?fct XAwuH))
"1"))))
; Get the number of verbs with sampraxAna AkAfkRA
(deffunction count-sampraxAna-AkAfkRA (?template)
(length (find-all-facts ((?fct ?template))
(eq (gdbm_lookup "sampraxAna_XAwu_list.gdbm" (fact-slot-value ?fct XAwuH))
"1"))))
; Get the number of verbs with apAxAna AkAfkRA
(deffunction count-apAxAna-AkAfkRA (?template)
(length (find-all-facts ((?fct ?template))
(eq (gdbm_lookup "apAxAna_XAwu_list.gdbm" (fact-slot-value ?fct XAwuH))
"1"))))
```

```
;Get the number of xviwIyA-upapaxa entries
(deffunction count-xviwIyA-upapaxa (?template)
(length (find-all-facts ((?fct ?template))
(eq (gdbm_lookup "xviwIyA_upapaxa_list.gdbm" (fact-slot-value ?fct word))
"1"))))

;Get the number of wqwIyA-upapaxa entries
(deffunction count-wqwIyA-upapaxa (?template)
(length (find-all-facts ((?fct ?template))
(eq (gdbm_lookup "wqwIyA_upapaxa_list.gdbm" (fact-slot-value ?fct rt)) "1"))))

;Get the number of cawurWI-upapaxa entries
(deffunction count-cawurWI-upapaxa (?template)
(length (find-all-facts ((?fct ?template))
(eq (gdbm_lookup "cawurWI_upapaxa_list.gdbm" (fact-slot-value ?fct word))
"1"))))

;Get the number of paFcamI-upapaxa entries
(deffunction count-paFcamI-upapaxa (?template)
(length (find-all-facts ((?fct ?template))
(eq (gdbm_lookup "paFcamI_upapaxa_list.gdbm" (fact-slot-value ?fct rt))
"1"))))

;Get the number of RaRTI-upapaxa entries
(deffunction count-RaRTI-upapaxa (?template)
(length (find-all-facts ((?fct ?template))
(eq (gdbm_lookup "RaRTI_upapaxa_list.gdbm" (fact-slot-value ?fct rt)) "1"))))

;Get the number of sapwamI-upapaxa entries
(deffunction count-sapwamI-upapaxa (?template)
```

```
(length (find-all-facts ((?fct ?template))
(eq (gdbm_lookup "sapwamI_upapaxa_list.gdbm" (fact-slot-value ?fct rt)) "1"))))

;check karmaNi
(deffunction karmaNi-vA (?template)
(any-factp ((?fct ?template))
(eq (fact-slot-value ?fct prayogaH) karmaNi)))

; Get the number of sup entries with praWamA
(deffunction count-xviwIyAnwa (?template)
(length (find-all-facts ((?fct ?template))
(= (fact-slot-value ?fct viBakwiH) 2))))

; Get the number of sup entries with wqwIyA
(deffunction count-wqwIyAnwa (?template)
(length (find-all-facts ((?fct ?template))
(= (fact-slot-value ?fct viBakwiH) 3))))

; Get the number of sup entries with cawurWI
(deffunction count-cawurWyAnwa (?template)
(length (find-all-facts ((?fct ?template))
(= (fact-slot-value ?fct viBakwiH) 4))))

; Get the number of sup entries with paFcamI
(deffunction count-paFcamyanwa (?template)
(length (find-all-facts ((?fct ?template))
(= (fact-slot-value ?fct viBakwiH) 5))))

; Get the number of sup entries with RaRTI
(deffunction count-RaRTanwa (?template)
```

```
(length (find-all-facts ((?fct ?template))
(= (fact-slot-value ?fct viBakwiH) 6))))

; Get the number of sup entries with sapwamI
(deffunction count-sapwamyanwa (?template)
(length (find-all-facts ((?fct ?template))
(= (fact-slot-value ?fct viBakwiH) 7))))

(open "bar.txt" bar "a")

;=============================================================
;rAmaH vexam paTawi
; rl1
(defrule assign_karwA_karma_karwqvAcy
(declare (salience 100))
(test (> (count-praWamAnwa sup) 0))
(test (> (count-xviwIyAnwa sup) 0))
(test (eq (karwari-vA wif) TRUE))
(or (test (= (count-xvikarmaka-AkAfkRA wif) 0)) (test (< (count-xviwIyAnwa
sup) 2)))
=>
(delayed-do-for-all-facts
((?s1 sup) (?s2 sup) (?w wif))
(and (= ?s1:viBakwiH 1) (= ?s2:viBakwiH 2) (<> ?s1:id ?s2:id ?w:id)(eq
?w:prayogaH karwari))
(modify ?w (karwA_pos ?s1:id) (karma_pos ?s2:id))
(modify ?s2 (kAraka karma))
(modify ?s1 (kAraka karwA))
(printout bar "(" ?s1:id " " ?w:word " kA karwA rl1)" crlf)
```

```
(printout bar "(" ?s2:id " " ?w:word " kA karma rl1)" crlf)
))
;;=============================================================
; rl2
(defrule assign_karwA_karwqvAcy
(declare (salience 100))
(test (> (count-praWamAnwa sup) 0))
(test (= (count-xvikarmaka-AkAfkRA wif) 0))
;(test (= (count-xviwIyAnwa sup) 0))
(test (eq (karwari-vA wif) TRUE))
=>
(delayed-do-for-all-facts
((?s1 sup) (?w wif))
(and (<> ?s1:id ?w:id) (= ?s1:viBakwiH 1) (eq ?w:prayogaH karwari))
(modify ?w (karwA_pos ?s1:id))
(modify ?s1 (kAraka karwA))
(printout bar "(" ?s1:id " " ?w:word " kA karwA rl2)" crlf)
)
)
;=============================================================
;rAmeNa vexaH paTyawe
;rl3
(defrule assign_karwA-karaNa-vA_karma_karmavAcy
(declare (salience 100))
(test (> (count-praWamAnwa sup) 0))
(test (> (count-wqwIyAnwa sup) 0))
```

```
;(any-factp (?w wif) (?w:prayogaH karmaNi))
(test (eq (karmaNi-vA wif) TRUE))
=>
(delayed-do-for-all-facts
((?s1 sup) (?s2 sup) (?w wif))
(and (= ?s1:viBakwiH 3) (= ?s2:viBakwiH 1) (<> ?s1:id ?s2:id ?w:id)(eq
?w:prayogaH karmaNi))
(modify ?w (karwA_pos ?s1:id) (karma_pos ?s2:id))
(modify ?s2 (kAraka karma))
(modify ?s1 (kAraka karwA))
(printout bar "(" ?s1:id " " ?w:word " kA karwA rl3)" crlf)
(printout bar "(" ?s2:id " " ?w:word " kA karma rl3)" crlf)
))
;;=============================================================
;xAwreNa lunAwi
;rl4
(defrule assign_karaNa_karwqvAcy
(declare (salience 100))
(test (> (count-wqwIyAnwa sup) 0))
(test (eq (karwari-vA wif) TRUE))
=>
(delayed-do-for-all-facts
((?s1 sup) (?w wif))
(and (<> ?s1:id ?w:id) (= ?s1:viBakwiH 3) (eq ?w:prayogaH karwari))
(modify ?w (karaNa_pos ?s1:id))
(modify ?s1 (kAraka karaNa))
(printout bar "(" ?s1:id " " ?w:word " kA karaNa rl4)" crlf)
```

```
)
)
;=============================================================
rAmeN bANena vAliH hanyawe
(defrule assign_karaNa_karmavAcy
(test (> (count-wqwIyAnwa sup) 0))
=>
(delayed-do-for-all-facts
((?s1 sup) (?s2 sup) (?s3 sup) (?w wif))
(and (= ?s1:viBakwiH 3) (= ?s2:viBakwiH 3) (= ?s3:viBakwiH 1) (<> ?s1:id
?s2:id ?s3:id ?w:id)(eq ?w:prayogaH karmaNi))
(modify ?w (karwA_pos ?s1:id) (karaNa_pos ?s2:id)(karma_pos ?s3:id))
(modify ?s1 (kAraka karwA))
(modify ?s2 (kAraka karaNa))
(modify ?s3 (kAraka karma))
(printout bar "(" ?s1:id " " ?w:word " kA karwA )" crlf)
(printout bar "(" ?s2:id " " ?w:word " kA karma )" crlf)
(printout bar "(" ?s3:id " " ?w:word " kA karaNa )" crlf)
))
;=============================================================
;rl5
(defrule assign_sampraxAna
(declare (salience 100))
(test (> (count-cawurWyAnwa sup) 0))
(or (test (> (count-xvikarmaka-AkAfkRA wif) 0)) (test (> (count-sampraxAna-
AkAfkRA wif) 0)))
=>
```

```
(delayed-do-for-all-facts
((?s1 sup) (?s2 sup) (?w wif))
(and (<> ?s1:id ?s2:id ?w:id) (= ?s1:viBakwiH 4) (or (eq (gdbm_lookup
"sampraxAna_XAwu_list.gdbm" ?w:XAwuH) "1") (eq (gdbm_lookup "xvikar-
maka_XAwu_list.gdbm" ?w:XAwuH) "1")))
;(= ?s1:viBakwiH 4)
(modify ?w (sampraxAna_pos ?s1:id))
(modify ?s1 (kAraka sampraxAna))
(printout bar "(" ?s1:id " " ?w:word " kA sampraxAna rl5)" crlf)
)
)

;=============================================================
;rl6
(defrule assign_apAxAna
(declare (salience 100))
(test (> (count-paFcamyanwa sup) 0))
=>
(delayed-do-for-all-facts
((?s1 sup) (?w wif))
(and (<> ?s1:id ?w:id) (= ?s1:viBakwiH 5))
(modify ?w (apAxAna_pos ?s1:id))
(modify ?s1 (kAraka apAxAna))
(printout bar "(" ?s1:id " " ?w:word " kA apAxAna rl6)" crlf)
)
)

;=============================================================
;rl7
```

```
(defrule assign_aXikaraNa
(declare (salience 100))
(test (> (count-sapwamyanwa sup) 0))
=>
(delayed-do-for-all-facts
((?s1 sup) (?w wif))
(and (= ?s1:viBakwiH 7) (<> ?s1:id ?w:id))
(modify ?w (aXikaraNa_pos ?s1:id))
(modify ?s1 (kAraka aXikaraNa))
(printout bar "(" ?s1:id " " ?w:word " kA aXikaraNa rl7)" crlf)
)
)
;=========================================================
;rl8
(defrule assign_xviwIyA-upapaxa
(test (> (count-xviwIyAnwa sup) 0))
(test (> (count-xviwIyA-upapaxa avy) 0))
=>
(delayed-do-for-all-facts
((?a avy) (?s sup))
(and (= ?s:viBakwiH 2) (= (- ?a:id ?s:id) 1) (eq (gdbm_lookup "xviwIyA_upapaxa_list.gdbm"
?a:word) "1"))
(modify ?s (kAraka xviwIyA-upapaxa_viBakwiH))
(printout bar "(" ?s:id " " ?a:word " kA xviwIyA-upapaxa_viBakwiH rl8)"
crlf)
)
)
```

```
;=============================================================
;rl9
(defrule assign_wqwIyA-upapaxa
(test (> (count-wqwIyAnwa sup) 0))
(test (> (count-wqwIyA-upapaxa avy sup) 0))
=>
(delayed-do-for-all-facts
((?a avy) (?s sup))
(and (= ?s:viBakwiH 3) (= (- ?a:id ?s:id) 1) (eq (gdbm_lookup "wqwIyA_upapaxa_list.gdbm"
?a:word) "1"))
(modify ?s (kAraka wqwIyA-upapaxa_viBakwiH))
(printout bar "(" ?s:id " " ?a:word " kA wqwIyA-upapaxa_viBakwiH rl9)"
crlf)))
;=============================================================
;rl10
(defrule assign_cawurWI-upapaxa
;(declare (salience 100))
(test (> (count-cawurWyAnwa sup) 0))
(test (> (count-cawurWI-upapaxa avy) 0))
=>
(delayed-do-for-all-facts
((?a avy) (?s sup))
(and (= ?s:viBakwiH 4) (= (- ?a:id ?s:id) 1) (eq (gdbm_lookup "cawurWI_upapaxa_list.gdbm
?a:word) "1"))
(modify ?s (kAraka cawurWI-upapaxa_viBakwiH))
(printout bar "(" ?s:id " " ?a:word " kA cawurWI-upapaxa_viBakwiH rl10)"
crlf)))
;=============================================================
```

```
;rl11
(defrule assign_paFcamI-upapaxa
(test (> (count-paFcamyanwa sup) 0))
(test (> (count-paFcamI-upapaxa avy sup) 0))
=>
(delayed-do-for-all-facts
((?a avy) (?s sup))
(and (= ?s:viBakwiH 5) (= (- ?a:id ?s:id) 1) (eq (gdbm_lookup "paFcamI_upapaxa_list.gdbm"
?a:word) "1"))
(modify ?s (kAraka paFcamI-upapaxa_viBakwiH))
(printout bar "(" ?s:id " " ?a:word " kA paFcamI-upapaxa_viBakwiH rl11)"
crlf)))
;=========================================================
;rl12
(defrule assign_RaRTI-upapaxa
(test (> (count-RaRTanwa sup) 0))
(test (> (count-RaRTI-upapaxa avy sup) 0))
=>
(delayed-do-for-all-facts
((?a avy) (?s sup))
(and (= ?s:viBakwiH 6) (= (- ?a:id ?s:id) 1) (eq (gdbm_lookup "RaRTI_upapaxa_list.gdbm"
?a:word) "1"))
(modify ?s (kAraka RaRTI-upapaxa_viBakwiH))
(printout bar "(" ?s:id " " ?a:word " kA RaRTI-upapaxa_viBakwiH rl12)"
crlf)))
;=========================================================
;rl13
(defrule assign_sapwamI-upapaxa
```

```
(test (> (count-sapwamyanwa sup) 0))

(test (> (count-sapwamI-upapaxa avy sup) 0))

=>

(delayed-do-for-all-facts

((?a avy) (?s sup))

(and (= ?s:viBakwiH 7) (= (- ?a:id ?s:id) 1) (eq (gdbm_lookup "sapwamI_upapaxa_list.gdbm
?a:word) "1"))

(modify ?s (kAraka sapwamI-upapaxa_viBakwiH))

(printout bar "(" ?s:id " " ?a:word " kA sapwamI-upapaxa_viBakwiH rl13)"
crlf)))

;=============================================================

;rl14

(defrule assign_xvikarmaka-karwqvAcy

(test (> (count-xvikarmaka-AkAfkRA wif) 0))

(test (> (count-xviwIyAnwa sup) 1))

(test (eq (karwari-vA wif) TRUE))

=>

(delayed-do-for-all-facts

((?s1 sup)(?s2 sup)(?s3 sup)(?w wif))

(and (<> ?s1:id ?s2:id ?s3:id ?w:id)(<> ?s2:id ?s3:id ?w:id) (<> ?s3:id
?w:id) (= ?s1:viBakwiH 1) (= ?s2:viBakwiH 2)(= ?s3:viBakwiH 2) (eq ?w:prayogaH
karwari)(eq (gdbm_lookup "xvikarmaka_XAwu_list.gdbm" ?w:XAwuH) "1"))

(modify ?w (karma_pos ?s2:id ?s3:id))

(modify ?s2 (kAraka gONa-muKya-karma-vA))

(modify ?s3 (kAraka gONa-muKya-karma-vA))

(printout bar "(" ?s3:id " " ?w:XAwuH " kA gONa-muKya-karma-vA rl14)"
crlf)

(printout bar "(" ?s1:id " " ?w:XAwuH " kA karwA rl14)" crlf)
```

```
(printout bar "(" ?s2:id " " ?w:XAwuH " kA gONa-muKya-karma-vA rl14)"
crlf)))

;=============================================================

;rl15

(defrule assign_xvikarmaka-karmavAcy

(test (> (count-xvikarmaka-AkAfkRA wif) 0))

(test (eq (karmaNi-vA wif) TRUE))

=>

(delayed-do-for-all-facts

((?s1 sup) (?s2 sup) (?s3 sup) (?w wif))

(and (<> ?s1:id ?s2:id ?w:id)(<> ?s2:id ?s3:id ?w:id)(<> ?s3:id ?w:id)(=
?s1:viBakwiH 3)(= ?s2:viBakwiH 1)(= ?s3:viBakwiH 2)(eq ?w:prayogaH kar-
maNi)(eq (gdbm_lookup "xvikarmaka_XAwu_list.gdbm" ?w:XAwuH) "1"))

(modify ?w (karma_pos ?s2:id ?s3:id))

(modify ?s2 (kAraka gONa-muKya-karma-vA))

(modify ?s3 (kAraka gONa-muKya-karma-vA))

(printout bar "(" ?s1:id " " ?w:XAwuH " kA karwA rl15)" crlf)

(printout bar "(" ?s3:id " " ?w:XAwuH " kA gONa-muKya-karma-vA rl15)"
crlf)

(printout bar "(" ?s2:id " " ?w:XAwuH " kA gONa-muKya-karma-vA rl15)"
crlf)))

;=============================================================

(agenda)

(run)

(facts)

(close bar)

(exit)
```

Morph Proune Rules

(reset)

;(facts)

; Get the number of sup entries with samboXana

(deffunction count-supkqw-samboXana (?template)

(length (find-all-facts ((?fct ?template))

(= (fact-slot-value ?fct viBakwiH) 8))))

; Check for the presence of sawi sapwami

(deffunction count-sawi-sapwami (?template)

(length (find-all-facts ((?fct ?template))

(and (eq (fact-slot-value ?fct viBakwiH) 7)(eq (fact-slot-value ?fct kqw_prawyayaH) Sawq)))))

; Check the presence of samboXana sUcaka avyayas

(deffunction count-samboXana-avy (?template)

(length (find-all-facts ((?fct ?template))

(eq (gdbm_lookup "samboXana_avy_wrds_list.gdbm" (fact-slot-value ?fct word)) "1"))))

;

; yasya ca BAvena BAvalakRaNam

(open "foo.txt" foo "a")

(open "for_kAraka.txt" bar "w")

(defrule sawi-sapwami

(test (>= (count-sawi-sapwami kqw) 1))

=>

; repeat for all

```
(do-for-all-facts

; kqxanwas
((?w kqw))

; having Sawq+7 form
(and
(eq ?w:kqw_prawyayaH Sawq)
(= ?w:viBakwiH 7)

; and no other word with 7 viBakwi in the sentence
(not (any-factp ((?s sup)) (= ?s:viBakwiH 7)))

; and the word under consideration has at least one non-Sawq+7 analysis
(any-factp ((?w1 sup wif)) (= ?w1:id ?w:id))
)

; then retract such analysis and also save this info in a file
(retract ?w)
(printout foo "(" ?w:id " " ?w:mid ") yasya ca BAvena BAvalakRaNam" crlf
)
)
)
;=============================================================

; rAma awra AgacCa
(defrule samboXana-1
(or (test (>= (count-supkqw-samboXana sup) 1))
(test (>= (count-supkqw-samboXana kqw) 1)))
=>
; For each of the wif sup pair
```

```
(do-for-all-facts
((?w wif)(?s sup kqw))

; If the wif is in lot or viXilif lakAra
; If the vacanam of the wif matches with the sup
; If the sup is in samboXana
; if the wif is not in uwwama puruRa
; If only one samboXana with these properties exists,

(and (<> ?w:id ?s:id) (or (eq ?w:lakAraH lot) (eq ?w:lakAraH viXilif)) (=
?s:viBakwiH 8) (eq ?w:vacanam ?s:vacanam) (neq ?w:puruRaH u) (or (=
(count-supkqw-samboXana sup) 1) (= (count-supkqw-samboXana kqw) 1)))

; Remove all non samboXana analysis of such a word in samboXana.
(do-for-all-facts
((?s1 sup wif kqw avy))
(and (<> ?s:mid ?s1:mid) (= ?s:id ?s1:id))
(retract ?s1)
(printout foo "(" ?s1:id " " ?s1:mid ") another analysis is removed " crlf )
)
)
)
;=============================================================
(defrule samboXana-2
(test (>= (count-samboXana-avy avy) 1))
(or (test (>= (count-supkqw-samboXana sup) 1)) (test (>= (count-supkqw-
samboXana kqw) 1)))

=>
; For each of the avy sup pair
```

```
(do-for-all-facts
((?a avy)(?s sup kqw))

; If the sup is in samboXana
; If the avy is from samboXana-avy

(and (<> ?a:id ?s:id) (gdbm_lookup "samboXana_avy_wrds_list.gdbm" ?a:word)
(= (count-supkqw-samboXana sup) 1) (= (count-supkqw-samboXana kqw)
1))

; Remove all non samboXana analysis of such a word in samboXana.
(do-for-all-facts
((?s1 sup wif kqw avy))
(and (<> ?s:mid ?s1:mid) (= ?s:id ?s1:id))
(retract ?s1)
(printout foo "(" ?s1:id " " ?s1:mid ") After the samboXana word samboXana
avyaya is there " crlf)
)
)
)
(run)
(facts)
(do-for-all-facts
((?s sup))
(printout bar "(sup (id " ?s:id ") (mid " ?s:mid ") (word " ?s:word ")
(rt " ?s:rt ")(lifgam "?s:lifgam ")(viBakwiH " ?s:viBakwiH ")(vacanam "
?s:vacanam ") )" crlf )
)
(do-for-all-facts
((?w wif))
```

```
(printout bar "(wif (id " ?w:id ") (mid " ?w:mid ") (word " ?w:word ") (rt
" ?w:rt ")(XAwuH " ?w:XAwuH ")(lakAraH " ?w:lakAraH ")(prayogaH "
?w:prayogaH ")(puruRaH " ?w:puruRaH ")(vacanam " ?w:vacanam ")(paxI
" ?w:paxI ")(gaNaH " ?w:gaNaH "))" crlf)

)

(do-for-all-facts

((?w kqw))

(printout bar "(kqw (id " ?w:id ") (mid " ?w:mid ") (word " ?w:word ")
(kqw_prawyayaH " ?w:kqw_prawyayaH ") (lifgam " ?w:lifgam ") (viBakwiH
" ?w:viBakwiH ") (vacanam " ?w:vacanam ") (rt " ?w:rt ") (XAwuH "
?w:XAwuH ") (gaNaH " ?w:gaNaH "))" crlf)

)

(do-for-all-facts

((?w avy))

(printout bar "(avy (id " ?w:id ") (mid " ?w:mid ") (word " ?w:word "))"
crlf)

)

(do-for-all-facts

((?w waxXiwa))

(printout bar "(waxXiwa (id " ?w:id ") (mid " ?w:mid ") (word " ?w:word ")
(rt " ?w:rt ")(lifgam " ?w:lifgam ")(viBakwiH " ?w:viBakwiH ")(vacanam"
?w:vacanam "))" crlf)

)

(close foo)

(close bar)

(exit)
```

LIST OF EXAMPLES:

पाचकः पचति
परशुना छिनत्ति
प्रासादात् पतति
पीठे उपविशति
पाचकः ओदनं पचति
पाचकः ओदनं पचते
पाचकेन तण्डुलः पच्यते
सूदः पचति
बालः खादति
वटुः लिखति
सूदः तण्डुलं पचति
बालकः ग्रामं गच्छति
चैत्रः कूपं खनति
भक्तः हरिं भजति
रामः बाणेन रावणं हन्ति
देवदत्तः परशुना वृक्षं छिनत्ति
बालः पादाभ्यां गृहं गच्छति
चैत्रः रजकाय वस्त्रं ददाति
राजा विप्राय गां ददाति
पर्णं वृक्षात् पतति
कृष्णः गोकुलात् आगच्छति
पान्थः पर्वतात् अवरोहति
खगः वृक्षात् डयते
बालः कटे उपविशति
सूदः ओदनं स्थाल्यां पचति
सूदः तण्डुलं पचति
सूदेन तण्डुलः पच्यते
गौः गोष्ठम् आगच्छति
छात्रेण श्लोकः पठ्यते
पाचकः तण्डुलान् पचति
भक्तः गङ्गां स्पृशति
गोपालः गां दोग्धि पयः
वामनः बलिं याचते वसुधाम्
पान्थः माणवकं पन्थानं पृच्छति

गुरुः बालं धर्मं ब्रूते
कृषीवलः अजां ग्रामं नयति
बालः विद्यालयं गच्छति
बालः गद्यं पठति
वृद्धः उपविशति
शिशुः शुकं पश्यति
बालः अन्नं खादति
बालः अन्नम् अत्ति
बालः अन्नं भक्षयति
कर्मकरः कटं करोति
अन्तरा त्वां मां हरिः
अन्तरेण हरिं न सुखम्
मार्गम् उभयतः वृक्षाः सन्ति
सर्वतः अध्यापकं शिष्याः सन्ति
धिक् अनृतवादिनम्
उपर्युपरि भूमिं विमानानि डयन्ते
गृहम् अभितः तृणानि सन्ति
विद्यालयं परितः छात्राः क्रिडन्ति
वृक्षं प्रति विद्योतते विद्युत्
अध्यापकेन विद्यालयः गम्यते
रामः बाणेन रावणं हन्ति
अक्षैः दीव्यति
बालः मित्रेण सह आगच्छति
अलं विस्तरेण
पिता पुत्राय फलं ददाति
बालकाय खण्डशर्करा रोचते
बालिका पुष्पेभ्यः स्पृहयति
कंसः कृष्णाय क्रुध्यति
शिशुः पन्थानं गच्छति
श्री गणेशाय नमः
अग्नये स्वाहा
पितृभ्यः स्वधा
वृक्षात् पर्णं पतति
पापात् जुगुप्सते
धर्मात् प्रमाद्यति

चोरात् बिभेति
व्याघ्रात् रक्षति
शिष्यः उपाध्यायात् अधीते
तन्तुभ्यः पटः भवति
मासात् आरभ्य मेघः वर्षति
तिलेषु तैलम् अस्ति
गुरुणा पाठः क्रियते
छात्राः शालां प्रविशन्ति
विश्वासः मित्राय पुस्तकं ददाति
बालः वाहनात् पतति
सर्वे कटे उपविशन्ति
मोक्षे इच्छा अस्ति
गोपेन गां व्रजः अवरुध्यते
पान्थेन माणवकः पन्थानं पृच्छ्यते
पित्रा माणवकः धर्मम् उच्यते
सुरेण सुधां क्षीरनिधिः मथ्यते
चैत्रेण देवदत्तः शतं मुष्यते
कृषकेण ग्रामम् अजा नीयते
गोपेन वृषभः गोष्ठं ह्रियते
कृषकेण अजा नगरम् उह्यते
वटुः वृक्षात् फलानि अवचिनोति
गोपः वृषभं गोष्ठे हरति
गोपालेन गौः पयः दुह्यते
वामनेन बलिः वसुधां याच्यते
भृत्यः भारं वहति
भृत्यः भारं नयति
कंसः कृष्णाय क्रुध्यति
सर्वेभ्यः स्वस्ति
ग्रामात् उत्तरः विद्यालयः
वैशाखात् पूर्वः चैत्रः
ग्रामात् दूरं नगरम्
ग्रामात् अन्तिकं गोष्ठम्
ग्रामस्य दूरात् वनम् अस्ति
वृक्षस्य निकटात् गौः अस्ति
ग्रामस्य दूरं वनम् अस्ति

ग्रामस्य दूरेण वनम् अस्ति
ग्रामस्य दूरात् वनम् अस्ति
वृक्षस्य अन्तिकात् गौः अस्ति
वृक्षस्य अन्तिकं गौः अस्ति
वृक्षस्य अन्तिकेन गौः अस्ति
वृक्षस्य अन्तिके गौः अस्ति
गृहस्य दूरं विद्यालयः
गृहात् दूरं विद्यालयः
विद्यालयस्य निकटम् आपणः
विद्यालयात् निकटम् आपणः
आलस्यात् न पठति
तक्रं विना कथं भोजनम्
तेन विना तृणमपि न चलति
ग्रामस्य अन्तिकं वृक्षः
ग्रामस्य दूरं वृक्षः
नगरस्य दूरेण ग्रामः
नगरस्य अन्तिकेन ग्रामः
गृहस्य दूरात् गोष्ठम् अस्ति
गृहस्य अन्तिकात् गोष्ठम् अस्ति
ग्रामस्य अन्तिके जलपातः
गृहात् दूरं विद्यालयः
गृहात् अन्तिकं विद्यालयः
गृहस्य दूरं विद्यालयः
गृहस्य अन्तिकं विद्यालयः
राज्ञा गर्गः शतं दण्ड्यते
पिता माणवकाय धर्मं ब्रूते
गुरुः माणवकाय धर्मं शास्ति
कृषिकः ग्रामं अजां नयति
गोपः गोवत्सं गोष्ठं कर्षति

Bibliography

- Subba Rao Veluri, 'The Philosophy of a sentene and its parts', Munshiram Manoharlal, New Delhi,1969
- Lokamani Dahal, परमलघुमञ्जुषा by नागेश भट्ट, Chaukhamba prakashan, Varanasi, 2006
- Charu Deva Shastri, Panini Re-interpreted, Motilal Banarasidas, Varanasi, 1990
- K V R Krishnacharyulu, भूषणसार समीक्षा, Raghavaprakasanam, Tirupati, 1996
- N. S. RAMANUJA TATACHARYA,शाब्दबोध मीमांसा, Institute Francis De Pondicherry, 2005.
- Brahmadatta Jijnasu, Panini Ashtadhyayi sutrapatha, Ramlal Kapoor Trust, Delhi, 1998
- Vedavratavarni, पतञ्जलि महाभाष्य, Harayana Sahitya samsthan, Rohataka, 2000
- J.G̃arratano and G.Riley, 'Expert Systems Principles and Programming', Course Technology, Delhi, 2007
- ईस्वरचन्द्र , अष्टाद्यायी, Chaukhamba Samskrita pratishtan, Delhi, 2004
- सत्यानन्दवेदवागीश , पाणिनीयशब्दानुशासनम्, BagvawI leser prints, New Delhi-65 विक्रम संवत्सरः 2060.
- विजयपाल विद्यावारिधिः , काशिका, Ramlal kapoor trust, Bagdad, Hariyana, 1997.
- आनन्दप्रकाश मेधार्थी , वार्तिक-प्रकाशः, caukhamba samskrita samsthaan, Varanasi, 1993.
- पण्डित सत्यानन्दवेदवागीश , संस्कृत-शिक्षण-सरणी , आचार्य राम शास्त्री ज्ञानपीठ, Samskrita nagar, RohinI, Delhi, 1983.