Independent Study: Computational Paninian Grammar **Spring 2013**

Lecture 1: January 21

Professor: Dr. Dipti Misra *Tags: CPG,Intro*

1.0.1 Introduction

The Karaka system serves as the basis for description of Panini's Syntax. It is a syntacito-semantic representation of the relations between the verb and the direct participants of the action in the sentence.

1.0.2 Definitions

Panini's work is explained, extended, commented and reinterpreted by many authors like kAtyayana, patanjali, bhartrhari and others. This section includes some of the definitions of the "kAraka".

- **Patanjali**, in his Mahabhashya defines "kAraka" as "karOti iti" ("The one that does")
- The author of **kAsika** explains it as being synonymous to "hEtu" and "nimitta" (Cause) - "kArakam hEtur ity anarthAntaram" (" Cause and kArakam are one and the same ")
- **Bhartrhari** uses the term "sAdhanam" to specify kAraka as the one capable of establishing action which is given the term "sAdhya".
- **NagEsa** defines kAraka as the one that producces the action.

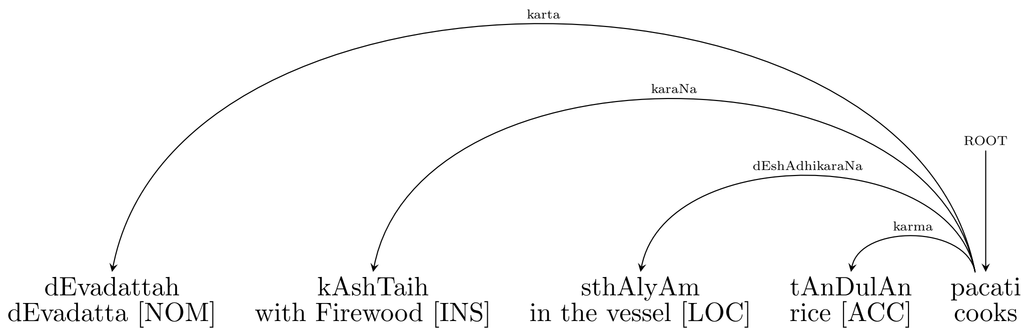
Therefore, we may say that kAraka is a animate/inanimate, passively/actively involved entity in the acomplishment of an action. The relations between the verb ("kriya") and the kAraka are of the type visheshaNa - visheshya (Modifier - Modified).

There are six kArakas. They are specified below briefly. (Written as per the order)

- **apAdAnam** : Defined as "dhruvam apAye pAdAnam" - The Entity which remains constant when seperation takes place
- **sampradAnam** : "karmaNA yam abhipraiti sa sampradAnam" - Is the entity for which the karma is intended.
- **karaNam** :
- **adhikaraNam**
- **karma**
- **karta**

1.0.3 Examples

Consider the following sentence.



" dEvadatta cooks the rice with the firewood in the vessel "

1.1 Some theorems and stuff

We now delve right into the proof.

Lemma 1.1 *This is the first lemma of the lecture.*

Proof: The proof is by induction on .... For fun, we throw in a figure.

Figure 1.1: A Fun Figure

This is the end of the proof, which is marked with a little box. ■

1.1.1 A few items of note

Here is an itemized list:

- this is the first item;
- this is the second item.

Here is an enumerated list:

1. this is the first item;
2. this is the second item.

Here is an exercise:

Exercise: Show that P $\neq$ NP.

Here is how to define things in the proper mathematical style. Let f_k be the $AND - OR$ function, defined by

$$f_k(x_1, x_2, \ldots, x_{2^k}) = \begin{cases} x_1 & \text{if } k = 0; \\ AND(f_{k-1}(x_1, \ldots, x_{2^{k-1}}), f_{k-1}(x_{2^{k-1}+1}, \ldots, x_{2^k})) & \text{if } k \text{ is even;} \\ OR(f_{k-1}(x_1, \ldots, x_{2^{k-1}}), f_{k-1}(x_{2^{k-1}+1}, \ldots, x_{2^k})) & \text{otherwise.} \end{cases}$$

Theorem 1.2 *This is the first theorem.*

Proof: This is the proof of the first theorem. We show how to write pseudo-code now.

Consider a comparison between x and y:

```
if x or y or both are in S then
    answer accordingly
else
    Make the element with the larger score (say x) win the comparison
    if F(x) + F(y) < n/(t-1) then
        F(x) ← F(x) + F(y)
        F(y) ← 0
    else
        S ← S ∪ {x}
        r ← r + 1
    endif
endif
```

This concludes the proof. ■

1.2 Next topic

Here is a citation, just for fun [CW87].

References

[CW87] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *Proceedings of the 19th ACM Symposium on Theory of Computing*, 1987, pp. 1–6.

Independent Study: Computational Paninian Grammar **Spring 2013**

Lecture 2: January 21

Professor: Dr. Dipti Misra *Tags: CPG,Intro*