1. Database and Data Separation in Astadhyayi – Its benefit

We had discussed here how data and algorithm are separated in Astadhyayi and what benefit does it bring?

Astadhyayi has around 4000 rules which is mentioned in the main body as Sutrapatha and beside that it has 5 ancillary texts: 14 maheswar sutras, ganpatha, dhatupatha, unadi sutras and linganusashanam. One benefit of keeping algorithm (sutrapatha) and data (ancillary text) which was discussed is maintainability: Which means that changes to data storage that is additions etc. during development or post development of Astadhyayi that is even if we want to add more words into any of the ancillary text can be made without affecting the logic and vice versa.

1. Data Encapsulation as discussed.

Data Encapsulation generally means to keep data and related methods/rules at same place as in a same Class, and it is often accompanied with data hiding or information hiding so that data should not be openly accessible and to modify or bring any change to it, only few methods can be utilized. In Astadhyayi this has been achieved by anubandha or *it* markers which is incorporated within the data itself which only triggers few rules and restrict other rules to apply.

1. Subroutines

Subroutines are methods or functions which takes some input and gives some output, so it has a certain function. We discussed one subroutine in Astadhyayi which identifies sounds used as markers(*it/anubandha*) in the text. It takes upadesha as input and gives all the markers(it/anubandha) as output. We had discussed 1.3.2 to 1.3.8 here along with Anubandha Concept and how this soubroutine can be written in a pseudo computer language using if-else statements.

1. Anubandha Concept

We had discussed here what is anubandha concept in Astadhyayi, drawn parallels to English grammar example if we have to use anubandha concept there, how we can use a affix to trigger a particular rule and thus the grammar can be kept compact and at the same time rules are incorporated.

1. With a simple sentence W1 W2 W3 V : Ramah gramam dwichakrena gachhati

We had seen the words can be written as W1 = R1 + S1, W2 = R2 + S2; W3 = R3 + S3, V = R4 + VS, where the root words R1, R2, R3, R4 relate to real world and denote concept but suffixes S1, S2, S3, S4 tells relation, question was if all the suffix together mark relation or alone VS is sufficient and can be taken a central pivot to do analysis:

where the information is coded, what is coded and how much is coded ?

1. We had seen the different levels of representation of speaker thoughts in Paninan Model: Semantic level -> karaka level -> Vibhakti level -> Surface level

Karaka level captures a certain level of semantic which is very close to semantic level and can help us in parsing the sentence. Vibhakti level however is purely syntactic.

Then we saw how we Astadhyai gives a mapping between karalka level and Vibahkti level and we can use this mapping to back trace the relationship and hence arrive at karaka information.

We had seen two sentences, one in active voice and one in passive voice and saw how same information is presented in different ways but the Karaka stiil remains the same and thus presenting a semantic structure for parsing.

We had also seen and design a parser based on this theory which can be given in following steps:

1. Find out associated grammar and morphological meaning for each word by simply looking into dictionary. We can also look for local word groupings here to mark noun, verb etc.
2. The vibhakti markers are identified and according to verb’s tense, aspect and modality, we can get a Karaka frame and transformation rules
3. A core parser then helps us to identify Karaka relations which requires karaka frame and transformation rules. For a given sentence, each of noun groups is tested against Karaka restrictions in the Karaka frame of the verb and if satisfied the word becomes a suitable candidate for the Karaka in the verb group.

Then we saw how a basic parsing output looks like and how words having relationship with each other pivot around the verb, depicted in a graph.