*Recent Research in Science and Technology 2011, 3(7): 109-111*
*ISSN: 2076-5061*
*www.scholarjournals.org*

www.recent-science.com

RRST-Computer Science

# Panini's Grammar in Computer Science

Parul Saxena[1*], Kuldeep Pandey[2], Vinay Saxena[2]

[1]*Department of Computer Science, Kumaun University, S.S.J. Campus Almora 263601, Uttarakhand, India*
[2]*Department of Mathematics, Kisan Post Graduate College, Bahraich 271801 (U.P.), India*

Article Info

Article History

*Received : 23-05-2011*
*Revised : 15-07-2011*
*Accepted : 21-07-2011*

*Corresponding Author

*Tel : +91-9415313670*
*Fax : (05962)232809*

*Email:*
*parul_saxena@yahoo.com*



Abstract

In the present work, we have described the contribution of the Great Indian mathematician Panini to the Modern Computer Science, it is the most appropriate approach of us to make the reader familiar with the great history of the Indian mathematics and modern applications of those. This will lead to new inventions and searches of the Indian mathematician to link with the modern world, which is the key aim of us.

Key Words: Panini's Grammar, Ashtadhyayi,Data and Database, Recursion, Array, Inheritance, Polymorphism, बहुब्रीहि समास, संधि

## Introduction

It seems a very amazing fact for learner that Panini, the founder of the Sanskrit Grammar, how may be a mathematician? About 100 years ago he was considered only a sage who establishes the Sanskrit Grammar in his major work ASHTADHYAYI or Astak[1-3]. The surprising facts had come into the existence, when Dr. P. Ramanujan (in 1989) presented a paper at 24th annual convention at computer society at Banglore in India that Sanskrit Grammar has something as a generator of the other languages like Chinese or Thai and this is the most perfect grammar in itself.

Now it is a reasonable question that what has Sanskrit grammar to do with Mathematics. Certainly this grammar has to do something with the modern Theoretical Computer Science or Mathematics or a computer programmer who can just recognize that how the theme of the modern programming language was drawn by an Indian saint before the 2400 years ago of the borning invention of the computers. ASHTADHYAYI, which is a collection of the grammar rules defining the structure and the syntaxes of the Sanskrit language. Since the programming languages are meant for the computers and to understand these, there should not be any ambiguity. The fundamentals (grammars) of these should be very clear and well establish with their rules and structures. This is the point where there are great similarities between the programming language and the Panini's grammar. The Panini's rules are based on the microanalysis of each syntactic format and there are many similarities between modern programming language application theories and Sanskrit grammar. A very few of them are described as follows:

## Concept of data and database:

For the computer by the Data we mean the raw facts and figures, which after arrangement, gives out the meaningful information and the collection of the many data is known as database. Data has not any meaning in itself but when it is arranged, it changes in to meaningful pattern. The databases used by panini are:

Aksharasam Amanaya(AS): There are fourteen sutras called the Shiva Sutras or the Maheshwar Sutras which work as a data to pronounciate any word.

Sutrapatha(SP): There are 4000 sutras in which 3983 are specially in kashikarath.

Datupata(DP): There are 1967 verb roots and if the kandvadi roots included then it becomes 2014 .

Ghanpatha(GP):These contain other pertinent items like primitive's nominal bases and avayayas.

AS, DP and GP are called as the Databases for the Sanskrit language.

## Recursion:

Panini used many conceptual techniques, which are alike same as the theories of the modern programming languages, recursion is one of them. Panini did not use all padas in each sutra to complete the meaning of the each sutra. For completeness he took some padas from previous sutras. Now there become a situation in which a sutra is called within itself and this process is called the recursion. For example:

उपदेषेस्त्यं हलिस्यात् उपदेष आधोच्चारणम्

सूत्रेढव दृष्टं पदं सूत्रान्तरादनुवर्तनीय सर्वत्र (1–3–3)

In this sutra there is a recursion of उपदेष. This term is defined in the another sutra

धातु सूत्र गणोणादि वाक्य लिङ.ानुषासनम् आगम प्रत्यय देषा उपदेषाः प्रकीर्तिता

Also used this term in the definition of the nasal word as follows-

उपदेषेऽजनुनासिक इत् (1–3–2)

Array:

Panini uses following 14 sutras known as Shiv sutras or Maheshwara sutras.

1. अइउण् 2.ऋलृक 3.एओंङ.: 4.ऐऔच 5.हयवरट 6.लण् 7. ञमङ.णनम् 8.झमञ् 9. घढधष् 10. जबगड्दश् 11.खफछठथचटतव् 12.कपय 13. शषसर् 14. हल्

These sutras have been used to make Pratyahara's (the useful sentences or statements). The last vowels of these formulas disappear as parallel to the same concept of the null character in character arrays or strings. Panini defined as follows-

अर्दानं लोपः प्रसक्त स्पादर्शनलोपसंज्ञं स्पात् (1–1–60)

तस्पेतो लोपः स्पात्णादयोऽणद्यर्याः (1–3–9)

This sutra means the non visual expression of any vowel is called लोपः . In the similar way a special character null is assigned which does not came into the existence while using it. For example:

| N | A | M | A | H | \0 | S | H | I | V | A | Y | \0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

Here in both strings Namah and Shivay there is a null character at last, which has no value in out put result. In the same way new line symbol '\n' and other and comment lines does not appears in the output. Hence by visualizing above facts we can just recognize that how the arrays are similar to the concept of the Maheshwar sutras. The concatenation of two strings was also defined by him as संधि as follows-

हलोऽनन्तरा संयोगः (1–1–7)

अञ्भि व्यवहिता हलः संयोगः संज्ञाः स्युः

Inheritance:

The similarity of a Panini's grammar rule with the concept of the inheritance is explained as follows. There is a class जवनिका which has two other base class line.

रामः अवस्तारों जवनिकाः । अवतारः कूपादेः ।(3–3–120)

The meaning of the word राम: is defined as another place as follows:

रमन्ते योगिनोस्मिन्नति रामः(3–3–121)

The statement inherits its complete meaning by the different three classes:

रामः, कूपः, अवनिका

It is very clear that this process consumes less space than the explanation of each term everywhere as parallel to a good programmer always wants to reduce the space complexity. Hence the use of these concepts shows a path towards inheritance. In C++ language it is called multiple inheritance, in which a derived class inherits its feature from other many classes.

Polymorphism:

Polymorphism is now a days one of the crucial feature of object oriented programming. It simply means one name having multiple meanings. Panini pronounced the similar approach, while declaring the बहुब्रीहि समास. In बहुब्रीहि समास, he explains a word having different meanings. He described it as follows: -

हलदन्तात सप्तम्याः संज्ञायाम् (6–3–9)

शेषो बहुब्रीहि (2–2–23)

Examples:

- **कण्ठे काल :**
  A man having black throat
  Black throat God or Shivji
- **उपछत पषुः रूदः**
  Person having donated animal called rudra
  Lord having animal vehicles or God Shankar
- **पीताम्बरः**
  Yellow color cloths
  God having yellow cloth or Vishnu ji
- **गजाननं**
  The mouth of the elephant.
  Lord having mouth of the elephant or Ganesh ji

It is a polymorphism and can be categorized as the run time polymorphism. Although in his examples a word has only two or three meanings but it shows that how the seeds of the concept were used by the Panini. Also the one word having different meaning implies that it was strong approach of him to give a module to the user to use his own opinion. So it is just similar as in modern programming languages to use a function according to the programmer's purpose.

According to the above justification Panini should be called the forerunner of the modern computer programming languages, as Kruth (1964) in a letter put a point that the meta syntactic notations should be renamed as "Panini-Backus Form" in place of "Backus Norm Form" (which is the general structure of the programming languages) . Mainly his purpose was to put attention that Backus was not first to use the form by which his name has been associated.

Panini grammar which is a collection of the 4000 algebraic rules and meta rules has been analyzed by scholars. Kak[4] reviewed Paninian approach to natural language processing (NLP) and compared it with the current knowledge representation system of the artificial intelligence and declared that Paninian generative rules and the metarules could assist further advances in NLP. The consistency of the system of Panini's rule, as tested by Fawler's Automation, were discussed by Staal[5]. The Panini's work looks like a mathematical treatise, so much so that Cardona [6-7] argues (in the orest of peacock) that "Algebraic reasoning". The Indian way of the representing number by words and ultimately true development of the modern number system in India are linked through the structure of Sanskrit language. The main attributes of these conclusions are to be illustrated as the work of the Ancient Indian logicians and the grammarians. Kak[4] , Staal [5],

Ingalls[8], Matilal [9], Brigs[10] suggested that many contemporary developments informal logic linguistics and the computer science are rediscovery of the work of the Indian master. But apart from the question of a current history of ideas, it rises the following important questions of significance of the Sanskritists as well as cognitive and other features of the grammar that yet not be rediscovered in the computer science remains to be seen. As the success of these aspects of its structure will have implication for the further advances in the computer science: knowledge representation and the linguistics. In particular, we are hopeful for the significant application in the natural language processing. The ongoing analysis of the structure of the Panini and those of later grammarians will be added by the development of the software to implement "*Ashtadhyayi*" in digital computer.

The specification of immediate interest to the computer scientist include analysis of the arrangement of rules and search for arrangements that are equivalent in terms of their generative power. The formal aspect of these arrangements and their relationship will help the notion of the distance between grammars, such a motion is immediate relevance for machine translation as well as researchers at National Aeronautical and space administration (NASA), looking Sanskrit as possible computer language because of its complete morphology that leaves a little room forever. Many Sanskritists hope that one day Sanskrit will become language of the world. It is a clear and precise language structure enhances communication in an open alternative mean for expression. The foundation of grammar of it is created a solid base for India's largely self-propagated path of progress in Philosophy, Law and Governance, Art theaters and Music Epistemology, Mathematics and Computer science.

The Panini's illustration of the operations, metarules, transformation and the other features make his grammar as path that lead toward the modern Turing machine. His rules generate many other grammar rich languages. Panini's theorem of the constraints and his grammar is devised forerunner to modern formal language theory (mathematical linguistics) and general grammar that work as a precursor for the competing. Panini's work was very economical in usage. The nature of every word has usages of great importance. Panini believes are like a man of scientific temper who believes in micro system than the macro system while using the word. We the men of today world should take inspired from him and follow a path, which lead toward such a glorious relation with humanity and become a model of good hopes to the coming generation.

## References


[1] Girish nath, 1993, *Morphology of case affixes a computational analysis*, M.Phil thesis, Jawahar Lal Nehru University New Dehli.

[2] Sharma Ram Nath, 1987, *The Astadhyayi of Panini*. Vol I, Munshiram Manoharlal Publishers Pvt. Ltd., New Delhi.

[3] Sharma Ram Nath, 1990, *The Astadhyayi of Panini*. Vol II, Munshiram Manoharlal Publishers Pvt. Ltd., New Delhi.

[4] S. Kak, 1987, "*The P_an. ini approach to natural language processing"*, Int. Journal of Approximate Reasoning, vol. 1, 117-130.

[5] Frits Staal, 1988, *Universals, Studies in Indian Logic and Linguistics*. Chicago:University of Chicago Press.

[6] Cardona, George, 1970, *Some Principles of Panini's Grammar*, Journal of Indian Philosophy, vol 1, 40-74.

[7] Cardona, George, 1965, *On translating and formalizing Paninian rules*. Journal of Oriental Institute, Baroda, vol 14, 306-14.

[8] D.H.H. Ingalls, 1951, *Materials for the Study of Navya-Ny_aya Logic.* Cambridge: Harvard University Press.

[9] B.K. Matilal, 1968, *The Navya-Ny_aya Doctrine of Negation.* Cambridge: Harvard University Press.

[10] R. Briggs, 1985, "*Knowledge representation in Sanskrit and arti_cial intelligence"*, AI Magazine, vol. 6, pp. 22-38.