Ancient Indian Knowledge

Shekhar Verma

IIIT Allahabad

Computer Science

- Computer science has two fundamental elements.

i. Development of techniques that make the elucidation of the computational structure of nature and the mind easier.

ii. The creation of new computing algorithms and machines that have powerful cognitive and computational abilities:

  - *This includes development of new techniques of representing and manipulating knowledge, inference and deduction.*

Why is this important?

- We can represent data and information through their attributes

Question:

- How to represent knowledge?

Why is this important?

Question:

- What are data, information and knowledge?

Panini's Grammar and Computer Science

- The tasks of representing and processing knowledge with a somewhat different emphasis has parallels in many ancient disciplines.

a) Grammarians have considered questions of relating facts about the physical world and cognition to linguistic expressions.

b) Logicians have developed formal structures to relate events and draw inferences from them.

Seen best in the work of ancient Indian logicians and grammarians.

Significant Question

Question

- Are there rules in ancient Indian logic and grammar that may be of use in making further advance in cognitive and computer sciences?

Answer

- Yes

- Grammars of natural language are as powerful as any computing machine.
- But since the setting of a grammar is so different from the typical purpose of a computer, this fact is often obscured.
- The formal structure of a grammar can be easily adapted so as to perform numerical processing.

- Certain rules of Panini's grammar, *Astadhyayi*, which are traditionally studied together with

a) The *dhatupatha* (a list of verbal roots arranged into sublists) and

b) The *ganapatha* (a list of various classes of morphs, one class being the *dhatupatha*)

- Do not merely deal with analysis of words (sabdanusasana) but in fact provide a structure for the analysis of sentences.

- Due to its algebraic nature and its comprehensiveness, the structure has been described as a machine generating words and sentences of Sanskrit.

- Composed in the succinct sutra style,

- *Astadhyayi* consists of nearly 4000 sutras that capture the fundamentals of Sanskrit language in terms of its phonology, morphology and syntax

- As in any formal system

- The structure consists of
  1. *Definitions*
  2. *theorems (linguistic facts), and*
  3. *meta-theorems (rules regarding rules).*

- The rules are of different kinds:
    1. *some are universal and context-sensitive transformations,*
    2. *others operate sequentially or recursively*

- These rules are expressed in three groups:

1. rules of interpretation or meta-rules (paribhasha rules)

2. rules of affixation (rules prescribing affixes after two kinds of basic dhatu roots)

3. rules of transformation for the stems and the suffixes (the morphophonemic rules).

Computer program

- A computer program has exactly the same general features of context-sensitive rules, recursion, and sequential rule application.

- It is not surprising, therefore, that these sutras have been compared to a computer program that generates Sanskrit sentences.

Panini's grammar is algebraic where a finite set of rules generates an infinite number of words and sentences.

- It is generally agreed that the Paninian system is based on a principle of economy, an Occam's razor.
- This makes the structure to be of special interest to cognitive scientists.

- Furthermore, development of logic has been seen as emerging from the background of grammatical categories

- Grammatical categories serve to express knowledge about the world.
- Panini's system of knowledge representation is based in the karaka theory.
- The karaka are deep structure relations that mediate mappings from semantic relations (such as agent, goal, location) to phonological representations (in terms of case-endings that may express voices) via surface structures (in terms of morphological categories such as nominal cases, prepositions, and verbal voices).

Karaka theory is of importance in natural language processing

- Problems of particular interest to the computer scientist include the arrangement of the rules and the smallest set of rules that would be equivalent to *Astadhyayi*.

- Rearranged rules, such as those by Bhattoj Diksita in his Siddhanta Kaumudi provide an invaluable frame of comparison.

- Panini's meta language has its own vocabulary, syntax, and grammar.

- An extensive use of abbreviated expressions and other devices has given it an appearance of a code language.

Observations

- Several formal features of Panini's grammar that have direct parallels in computer science.

- Knowledge Representation is unique

Hashing & The Katapayadi Scheme

- Hashing is used in Databases, Cryptographic tools and other areas like Blockchain (that powers Crypto currencies like Bitcoin)
- The Katapayadi Scheme is a system of expressing numbers through the use of letters (Sanskrit consonants in this case), with more than one synonym for each number.
- .

- In the 18th century, we find Katapayadi Scheme in South Indian musicology, which is arguably similar to modern hashing

- Katapayadi Scheme can be treated as an early precursor to the modern hash functions, and its application in South Indian musicology bears, in retrospect, an interesting similarity to modem hash tables

Panini-Backus Form in Syntax of Formal Languages

- Vyaas Houston (1991), in one of his writings, mentions his discovery of the world's oldest living language: Sanskrit, the language of ancient India and Vedic civilization.

- He states thus:

"It was perfectly clear to me that I had come upon a perfect language, a language that invokes the spirit, an inexhaustible wellspring of spiritual inspiration. Where did it come from? — A language infinitely more sophisticated than any of our modem tongues."

Panini-Backus Form in Syntax of Formal Languages

- The sophistication refers to is about the formalism and structure of the language.

- For computer scientists, in the theory of formal languages, the word "formal" refers to the fact that all the rules for the language are explicitly stated in terms of what strings of symbols could occur, without any ambiguity and the need for interpretations based on mental skill

- Sanskrit not only has a very rich inflectional structure but this fact was recognized early by grammarians.

- The formal structure of computer programming languages was introduced in the 1958-60 period by John Backus (1958), and Peter Naur (1963).

- They headed UNESCO conferences on International algorithmic language ALGOL 60, a language "suitable for expressing a large class of numerical processes in a form sufficiently concise for direct automatic translation into the language of programmable automatic computers."

- BNF is an acronym originally for Backus-Naur Form.
- BNF notation can be found in any book on programming languages.

- In order to describe the (rather complicated) rules of grammar,

- Panini invented a notation which is equivalent in its power to that of Backus, and has many similar properties:

- Given the use to which the notation was put, it is possible to identify structures equivalent to the Backus "I" and to the use of the meta-brackets "<“ and ">" enclosing suggestive names.

Grammar

- Panini’s grammar (6th century B.C.E. or earlier) provides 4,000 rules that describe the Sanskrit of his day completely.
- Panini set out to describe the entire grammar in terms of a finite number of rules.
- Frits Staal (1988) has shown that the grammar of Panini represents a universal grammatical and computing system.
- From this perspective it anticipates the logical framework of modern computers (Kak 1987).

Binary numbers

- Barend van Nooten (1993) has shown that binary numbers were known at the time of Pingala's Chhandahshastra.
- Pingala, who lived around the early first century B.C.E., used binary numbers to classify Vedic meters.
- The knowledge of binary numbers indicates a deep understanding of arithmetic.

The Shri Yantra

- The Shri Yantra consists of nine triangles inscribed within a circle which leads to the formation of 43 little triangles.

- The yantra was made both on flat and curved surfaces during the middle ages.

- The drawing of the triangles on the curved surface implies the knowledge that sum of the angles of such triangles exceeds 180 degrees.
  - *Indicates a knowledge of non-Euclidean geometry*

P versus NP Problem

- Suppose we have a large group of students that we need to pair up to work on projects.

- We know which students are compatible with each other and we want to put them in compatible groups of two.

- We could search all possible pairings but even for 40 students we would have more than three hundred billion trillion possible pairings.

P versus NP Problem

- In 1965, Jack Edmonds gave an efficient algorithm to solve this matching problem

- Suggested a formal definition of “efficient computation" (runs in time a fixed polynomial of the input size).

- The class of problems with efficient solutions - known as P for “Polynomial Time".

P versus NP Problem

- But many related problems do not seem to have such an efficient algorithm.

- What if we wanted to make groups of three students with each pair of students in each group compatible (Partition Into Triangles)?

- What if we wanted to find a large group of students all of whom are compatible with each other (Clique)?

P versus NP Problem

- What if we wanted to sit the students around a large round table with no incompatible students sitting next to each other (Hamiltonian Cycle)?

- What if we put the students into three groups so that each student is in the same group with only his or her compatibles (3-Coloring)?

P versus NP Problem

- All these problems have a similar flavor:

- Given a potential solution, for example a seating chart for the round table, we can validate that solution efficiently.

- The collection of problems that have efficiently verifiable solutions is known as NP (for “Nondeterministic Polynomial-Time" if you have to ask)

Some NP Complete Problems

- Finding a DNA sequence that best fits a collection of fragments of the sequence
- Prime number factorization
- ….

Using Hardness

- Public key cryptography depends on this (For example: RSA Algorithm is based on the hardness of prime number factorization)

- Prime number factorization is hard to perform, easy to verify

- The math involved is modulo and requires finding the GCD
  - *GCD seemingly requires prime number factorization*

Using Hardness

- Suppose Alice's husband Bob is working on a Sudoku puzzle and Alice claims she has a solution to the puzzle
  - *solving a* $n^2 \times n^2$ *-Sudoku puzzle is NP-complete).*

- Can Alice convince Bob that she knows a solution without revealing any piece of it?

- Requires zero knowledge proof

Computer Science

- Uses hardness for cryptography, which is used for security.

- Requires formal proof methods

- Did we have hints of this in our ancient knowledge

- Yes

Aryabhata Remainder Theorem
Relevance To Public-key Crypto-algorithms

- Public-key crypto-algorithms are widely employed for authentication, signatures, secret-key generation and access control

- An algorithm from Aryabhatiya solves the indeterminate equation

$$a \cdot x + c = b \cdot y$$

- of degree one (also known as the Diophantine equation)

Aryabhata Remainder Theorem
Relevance To Public-key Crypto-algorithms

- Extension to solve the system of two residues

$$X \bmod m_i = X_i \ (for\ i = 1,2)$$

- This contribution known as the Aryabhatiya algorithm is very profound in the sense that the problem of two congruences was solved with just one modular inverse operation and a modular reduction to a smaller modulus than the compound modulus.

Kuṭṭaka

- An algorithm for finding integer solutions of linear Diophantine equations

- A linear Diophantine equation is an equation of the form

$$ax + by = c$$

where x and y are unknown quantities and a, b, and c are known quantities with integer values.

Kuṭṭaka

- The algorithm was originally invented by the Indian astronomer-mathematician Āryabhaṭa (476–550 CE) and is described very briefly in his Āryabhaṭīya.

- Bhāskara I (c. 600 – c. 680) who gave a detailed description of the algorithm with several examples from astronomy in his *Āryabhatiyabhāṣya*, who gave the algorithm the name *Kuṭṭaka*.

अधिकाग्रभागहारं छिन्द्यादूनाग्रभागहारेण ।
शेषपरस्परभक्तं मतिगुणमग्रान्तरे क्षिप्तं ॥
अधउपरिगुणितमन्त्ययुगूनाग्रच्छेदभाजिते शेषं ।
मधिकाग्रच्छेदगुणं द्विच्छेदाग्रमधिकाग्रयुतम् ॥

Kuṭṭaka

- The Kuṭṭaka algorithm has similarity with and is a precursor of the extended Euclidean algorithm.

- The algorithm is a procedure for finding integers x and y satisfying the condition

$$ax + by = \gcd(a, b)$$

Aryabhata Remainder theorem

- The improved Aryabhata algorithm is used to determine the multiplicative inverse $a^{-1}\text{mod}b$ by solving the equation $ax + by = 1$

- An iterative algorithm is also given to solve for

$$t \text{ moduli} m_i \ \ (i \ = \ 1,2, \ldots, t)$$

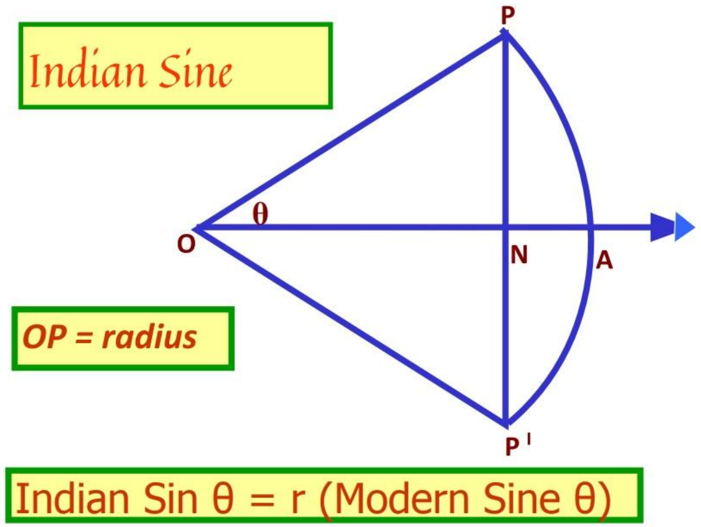
- The ART, which has much in common with the extended Euclidean algorithm, Chinese remainder theorem (CRT) and Garner's algorithm (GA), is shown to have a complexity comparable to or better than that of the CRT and GA

Āryabhaṭa's sine table

- Composed for the computation of the half-chords of certain set of arcs of a circle.

- It is a table of the first differences of the values of trigonometric sines expressed in arcminutes.

  - *In the Indian mathematical tradition, the sine (or jya) of an angle is not a ratio of numbers.*
  - *It is the length of a certain line segment, a certain half-chord.*



Āryabhaṭa's sine table

- Stanza in Āryabhaṭiya describing the sine table

मखि भखि फखि धखि णखि ञखि ङखि हस्झ स्ककि किष्ग श्घकि किघ्व ।
घ्लकि किग्र हक्य धकि किच स्ग झश ङ्व क्ल प्त फ छ कला-अर्ध-ज्यास् ॥

- The values encoded in Āryabhaṭa's Sanskrit verse can be decoded using the numerical scheme explained in Āryabhaṭīya, and the decoded numbers are listed in the table.

Āryabhaṭa's sine table

- In the table, the angle measures relevant to Āryabhaṭa's sine table are listed in the second column.
- The third column contains the list the numbers contained in the Sanskrit verse given above in Devanagari script.

Aryabhatta used value of R=3438.

Jya (Jiva) refers ardha-Jya (Jiva) *(adrha dropped)*

Ardha Jya = RSinθ
Sinθ = R/ardha Jya.
*Since R is 3438 (from 360*60/2 pi)*
& changing ardha Jya can be known when bow bends, Sine Table can be easily Computed

Example :

so *jyā* 45degrees = R sin (45degree) = 3438 x 0.7071 = 2431

See how closely values are matching with Modern mathematics

Sl. No	Angle (A) (in degrees, arcminutes)	Value in Āryabhaṭa's numerical notation (in Devanagari)	Value in Āryabhaṭa's numerical notation (in ISO 15919 transliteration)	Value in Hindu-Arabic numerals	Āryabhaṭa's value of jya (A)	Modern value of jya (A) (3438 × sin (A))
1	03° 45′	मखि	makhi	225	225′	224.8560
2	07° 30′	भखि	bhakhi	224	449′	448.7490
3	11° 15′	फखि	phakhi	222	671′	670.7205
4	15° 00′	धखि	dhakhi	219	890′	889.8199
5	18° 45′	णखि	ṇakhi	215	1105′	1105.1089
6	22° 30′	ञखि	ñakhi	210	1315′	1315.6656
7	26° 15′	ङखि	ṅakhi	205	1520′	1520.5885
8	30° 00′	हस्झ	hasjha	199	1719′	1719.0000
9	33° 45′	स्ककि	skaki	191	1910′	1910.0505
10	37° 30′	किष्ग	kiṣga	183	2093′	2092.9218
11	41° 15′	श्घकि	śghaki	174	2267′	2266.8309
12	45° 00′	किघ्व	kighva	164	2431′	2431.0331
13	48° 45′	घ्लकि	ghlaki	153	2585′	2584.8253
14	52° 30′	किग्र	kigra	143	2728′	2727.5488
15	56° 15′	हक्य	hakya	131	2859′	2858.5925
16	60° 00′	धकि	dhaki	119	2978′	2977.3953
17	63° 45′	किच	kica	106	3084′	3083.4485
18	67° 30′	स्ग	sga	93	3177′	3176.2978
19	71° 15′	झश	jhaśa	79	3256′	3255.5458
20	75° 00′	ङ्व	ṅva	65	3321′	3320.8530
21	78° 45′	क्ल	kla	51	3372′	3371.9398
22	82° 30′	प्त	pta	37	3409′	3408.5874
23	86° 15′	फ	pha	22	3431′	3430.6390
24	90° 00′	छ	cha	7	3438′	3438.0000

***jyā* 45degrees**

sum of ***jya***
225+224+...164
= **2431**

Mahabhaskariya (Book)

- Bhaskara suggested an approximate value of sine, which was unbelievably equal to accurate value of sine. The formula is given by:-

$$\sin x = 16x(p-x)/5p^2 - 4x\,(p-x).$$

- The approximate value of p was considered to be 10 for many centuries.
  - *But, according to Bhaskara, p had an irrational value, which was proved to be true later.*

Multiple Simultaneous Equations

- The sutras namely, Lopa- Sthapana Sutra, the Anurupya Sutra and the Paravartya sutra are used for solving the simultaneous equations.

First type

- A significant figure on the R.H.S of only one equation and zeroes on the other two equations.

$$x + y - z = 0 \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots. (A)$$
$$4x - 5y + 2z = 0 \quad \dots\dots\dots\dots\dots\dots\dots (B)$$
$$3x + 5y + z = 10 \quad \dots\dots\dots\dots\dots\dots\dots (C)$$

- We derive a new equation from the homogeneous zero equation, defining any two of the unknowns in terms of the third unknown.

- We then substitute the values in the third equation.

Jain mathematics

- Divided the large numbers and infinites into three categories:- enumerable, innumerable and infinite.

- They also described five types of infinity as:
  - *the infinite in one direction,*
  - *the infinite in two direction,*
  - *the infinite in area,*
  - *the infinite everywhere ,and*
  - *the infinite perpetually.*

amhitas and Brahmanas – use of large numbers

- In Yajurvedasaṃhitā- (1200–900 BCE), numbers as high as 10^{12} were being included in the texts. For example
  - *the mantra (sacred recitation) at the end of the annahoma ("food-oblation rite") performed during the aśvamedha, and uttered just before-, during-, and just after sunrise, invokes powers of ten from a hundred to a trillion:*
    - Hail to śata ("hundred," 10^2),
    - hail to sahasra ("thousand," 10^3),
    - hail to ayuta ("ten thousand," 10^4),
    - …
    - hail to parārdha ("one trillion," 10^{12} lit., "beyond parts"),
    - hail to the dawn (uṣas), hail to the twilight (vyuṣṭi),
    - …,
    - hail to svarga (the heaven),
    - hail to martya (the world), hail to all.

Proofs

- Proof is the primary vehicle for knowledge generation in mathematics.

- In computer science, proof has found an additional use:
  - *verifying that a particular system (or component, or algorithm) has certain desirable properties.*

Proofs

- One can practice computer science and do it successfully without ever having to deal with proofs explicitly.

- But computer science practitioners need to reason all the time: about algorithms, code, distributed systems, security protocols, and so on.

- To reason about complicated systems correctly and fluently, it is vital– to be familiar with fundamental proof techniques, because these techniques capture the most expressive and sound reasoning idioms, patterns, and laws.

Krsna Daivaj~na on the Importance of Upapatti

The following passage from Kṛṣṇa Daivajña's commentary on *Bījagaṇita* brings out the general understanding of the Indian mathematicians that citing any number of favourable instances (even an infinite number of them) where a result seems to hold, does not amount to establishing it as a valid result in mathematics. Only when the result is supported by an *upapatti* or demonstration can the result be accepted as valid.

ननूपपत्त्या विना वर्गयोगो द्विघ्नघातेन युतो हीनो वा युतिवर्गोऽन्तरवर्गो वा भवतीत्येतदेव कथम्? क्वचिद्दर्शनं त्वप्रयोजकम्। अन्यथा चतुर्गुणो राशिघातो युतिवर्गो भवतीत्यपि सुवचम्। तस्यापि क्वचित्तथा दर्शनात्। तथाहि राशी २, २ अनयोर्घातः ४ चतुर्गुणः १६ अयं जातो युतिः ४ वर्गः १६ वा राशी ३,३ अनयोर्घातश्चतुर्गुणः ३६ अयमेव युति ६ वर्गश्च ३६ वा राशी ४, ४ अनयोर्घातः १६ चतुर्गुणः ६४ अयमेव युति ८ वर्गः ६४ इत्यादिषु। तस्मात् क्वचिद्दर्शनम् अप्रयोजकं क्वचिद्व्यभिचारस्यापि संभवात्। अतो वर्गयोगो द्विघ्नघातयुतोनो युतिवर्गोऽन्तरवर्गश्च भवतीत्यत्र युक्तिर्वक्तव्येति चेत् सत्यम्। इयमुपपत्तिरेकवर्णमध्यमाहरणान्ते।

Bhaskara on Upapatti

In his discussion of Aryabhaṭa's approximate value of the ratio of the circumference and diameter of a circle, Bhāskara I notes that the approximate value is given, as the exact value cannot be given. He then goes on to argue that other values which have been proposed are without any justification:

एवं मन्यन्ते स उपाय एव नास्ति येन सूक्ष्मपरिधिरानीयते। ननु चायमस्ति

> विक्खंभवग्गदसगुणकरणी वट्टस्स परिरओ होवि।
> (विष्कम्भवर्गदशगुणकरणी वृत्तस्यपरिणाहो भवति)

इति। अत्रापि केवल एवागमः नैवोपपत्तिः। रूपविष्कम्भस्य दशकरण्यः परिधिरिति। अथ मन्यन्ते प्रत्यक्षेणैव प्रमीयमाणो रूपविष्कम्भक्षेत्रस्य परिधिर्दशकरण्य इति। नैतत् अपरिभाषितप्रमाणत्वात् करणीनाम्। एकत्रिविस्तारायामायतचतुरश्रक्षेत्रकर्णेन दशकरणिकेनैव तद्विष्कम्भ-परिधिर्वेष्ट्यमाणः स तत्प्रमाणो भवतीति चेत्तदपि साध्यमेव।

In the *madhyamāharāṇa* section Bhāskara poses the following problem

> क्षेत्रे तिथिनखैस्तुल्ये दोः कोटी तत्र का श्रुतिः।
> उपपत्तिश्च रूढस्य गणितस्यास्य कथ्यताम्।
>
> "In a right angled triangle with sides 15 and 20 what is the hypotenuse? Also give the demonstration for this traditional method of calculation."

Here Bhāskara gives two proofs. First the geometrical:

अत्र कर्णः या १। एतत् त्र्यस्रं परिवर्त्य यावत्तावत्कर्णो भूः कल्पिता। भुजकोटी तु भुजौ तत्र यो लम्बस्तदुभयतो ये त्र्यस्रे तयोरपि भुजकोटी पूर्वरूपे भवतः। अतस्त्रैराशिकं यदि यावत्तावति कर्णेऽयं १५ भुजस्तदा भुजतुल्ये कर्णे क इति लब्धो भुजः स्यात्। सा भुजाश्रिताऽऽबाधा २२५/या। पुनर्यदि यावत्तावति कर्ण इयं २० कोटिस्तदा कोटितुल्ये कर्णे केति जाता कोट्याश्रिताबाधा ४००/या। आबाधायुतिर्यावत्तावत्कर्णसमा क्रियते तावद्भुजकोटिवर्गयोगस्य पदं कर्णमानमुपपद्यते। अनेनोत्थापिते जाते आबाधे ९, १६ ततो लम्बः १२। क्षेत्रदर्शनम्।

Krsna Daivajna's Upapatti of Kuttaka Process

The *kuṭṭaka* procedure is for solving first order indeterminate equations of the form

$$\frac{(ax + c)}{b} = y$$

Here, **a, b, c** are given integers (called *bhājya, bhājaka* and *kṣepa*) and **x, y** are to be solved for in integers.

Kṛṣṇa first shows that the solutions for **x, y** do not vary if we factor all three numbers **a, b, c** by the same common factor.

He then shows that if **a** and **b** have a common factor then the above equation will not have a solution unless **c** is also divisible by the same.

He then gives the *upapatti* for the process of finding the *apavartāṅka* (greatest common divisor) of **a** and **b** by mutual division (the so-called Euclidean algorithm).

Krsna Daivajna's Upapatti of Kuttaka Process

Kṛṣṇa then provides a detailed justification for the *kuṭṭaka* method of finding the solution by making a *vallī* (table) of the quotients obtained in the above mutual division, based on a detailed analysis of the various operations in reverse (*vyasta-vidhi*).

In doing the reverse computation on the *vallī* (*vallyupasaṃhāra*) the numbers obtained, at each stage, are shown to be the solutions to the *kuṭṭaka* problem for the successive pairs of remainders (taken in reverse order from the end) which arise in the mutual division of **a** and **b**.

After analysing the reverse process of computation with the *vallī*, Kṛṣṇa shows how the solutions thus obtained are for positive and negative *kṣepa*, depending upon whether there are odd or even number of coefficients generated in the above mutual division.

And this indeed leads to the different procedures to be adopted for solving the equation depending on whether there are odd or even number of quotients in the mutual division.

Use of Tarka in Upapatti

The method of "proof by contradiction" is referred to as *tarka* in Indian logic. We see that this method is employed in order to show the non-existence of an entity.

For instance, Kṛṣṇa Daivajña essentially employs *tarka* to show the non-existence of the square-root of a negative number while commenting on the statement of Bhāskara that a negative number has no root.

वर्गस्य हि मूलं लभ्यते। ऋणाङ्कस्तु न वर्गः कथमतस्तस्य मूलं लभ्यते। ननु ऋणाङ्कः कुतो वर्गो न भवति न हि राजनिर्देशः।... सत्यम्। ऋणाङ्कं वर्गं वदता भवता कस्य स वर्ग इति वक्तव्यम्। न तावद्धनाङ्कस्य "समद्विघातो हि वर्गः" तत्र धनाङ्केन धनाङ्के गुणिते यो वर्गो भवेत् स धनमेव "स्वयोर्वधः स्वम्" इत्युक्तत्वात्। नाप्यृणाङ्कस्य। तत्रापि समद्विघातार्थमृणाङ्केनर्णाङ्कगुणिते धनमेव वर्गो भवेत् "अस्वयोर्वधः स्वम्" इत्युक्तत्वात्। एवं सति कथमपि तमङ्कं न पश्यामो यस्य वर्गः क्षयो भवेत्।

Use of Tarka in Upapatti

Thus according to Kṛṣṇa

> "The square-root can be obtained only for a square. A negative number is not a square. Hence how can we consider its square-root? It might however be argued: 'Why will a negative number not be a square? Surely it is not a royal fiat'... Agreed. Let it be stated by you who claim that a negative number is a square as to whose square it is; surely not of a positive number, for the square of a positive number is always positive by the rule... not also of a negative number. Because then also the square will be positive by the rule... This being the case, we do not see any such number whose square becomes negative..."

Use of Tarka in Upapatti

While the method of "proof by contradiction" or *reductio ad absurdum* has been used to show the non-existence of entities, the Indian mathematicians do not use this method to show the existence of entities, whose existence cannot be demonstrated by other direct means. They have a "constructive approach" to the issue of mathematical existence.

It is a general principle of Indian logic that *tarka* is not accepted as an independent *pramāṇa*, but only as an aid to other *pramāṇas*.

Indian Logic Excludes Aprasiddha Entities from Logical Discourse

Naiyāyikas or Indian logicians do not grant any scheme of inference, where a premise which is known to be false is used to arrive at a conclusion, the status of an independent *pramāṇa* or means of gaining valid knowledge.

In fact, they go much further in exorcising the logical discourse of all *aprasiddha* terms or terms such as “rabbit’s horn” (*śaśaśṛṅga*) which are empty, non-denoting or unsubstantiated.

Rhythms of life

- Vedas speak of the connections between the external and the internal worlds.

- The hymns speak often of the stars and the planets.
  - *These are sometimes the luminaries in the sky, or those in the firmament of our inner landscapes or both*

Fire, having become speech, entered the mouth
Air, becoming scent, entered the nostrils
The sun, becoming sight, entered the eyes
The regions becoming hearing, entered the ears
The plants, becoming hairs, entered the skin
The moon, having become mind, entered the heart.

—Aitreya Aranyaka 2.4.2.4

- This verse from the Upanishadic period speaks at many levels.
- At the literal level there is an association of the elements with various cognitive centers.
- At another level, the verse connects the time evolution of the external object to the cognitive center.

That’s All