The Ancient Katapayadi Formula And The Modern Hashing Method

Anand Raman
Department of Computer Science
Massey University
Palmerston North
New Zealand
A.Raman@massey.ac.nz

Abstract

The essence of the modern hashing technique in Computer Science is the derivation of a number from a non-numeric key to index into a table where the record containing the key is stored. It is believed that the idea of hashing was first seriously considered by H.P.Luhn of IBM in 1953. In this paper, an interestingly similar technique used in South Indian musicology in the 18th century is described and the question of whether it is an anticipation of the Hashing technique is briefly addressed.

1 Introduction

The problem of retrieving a record from a table based upon a given key has been studied extensively. A survey of work in this area can be found in (Severance 1974). In this paper I describe one particular approach to this problem — Hashing, and also an interesting earlier development very similar to it. It is generally believed that the idea of hashing was originated by H.P.Luhn, in an internal IBM memorandum in 1953 (Knuth 1973) and described first in the open literature by Arnold Dumey (Dumey 1956). But is it possible that the Katapayadi scheme of deriving numbers from names, in conjunction with the applications to which it had been put, especially in classical South Indian musicology, is an early anticipation of the hashing technique? We will look at this issue in more detail here.

2 Hashing

A hash-table is a data structure in which it takes on average constant time to find any given element. This constant time is the time taken to compute a function called the hash-function of the element being searched for. This is in contrast to a Binary Search Tree data structure, for example, in which the time taken to find an element is on average proportional to the $\log_2 N$, an array or linked linear list data structures in which the time is proportional to N where N is the total number of elements. The following example illustrates the use of hashing where the marks of ten students need to be stored in a table. It is a trivial one, but it is sufficient to bring out the essential principle behind hashing.

2.1 Example

Examination marks for ten students Amy, Ben, Cho, Dan, Eva, Fan, Gus, Hal, Ianand Jimneed to be stored in a table. We might additionally want to retrieve the mark of a student on demand, and optionally modify it. One way of doing this is to store the marks sequentially in a table of size 10, and perform a sequential search on it each time we want to retrieve a particular record. This would mean that on average, we can expect to scan half the table (5 elements) before finding the desired record. A more efficient storage technique will be to store the elements in sorted order by name. In this case, we would expect to search the table $\log_2 10$ (approximately 3.2) times on average for each retrieval, because at each examination, our search space is effectively halved as the element we want is either current, in the upper half or lower half depending on whether it is equal to, less than or greater than the current element.

In contrast to these techniques, the hashing scheme derives a unique number corresponding to each name which gives us the cell address of the element in the table. If we used a hash function H(x) = (ascii(x[0])-5) % 10 + 1, where x is the name or value being hashed, x[0] is the first letter of that name, ascii() is a function that returns the ASCII value of a given letter, and % stood for the modulus or remainder operator, then the following arrangement of elements in the table would be seen.

Addr	0	1	2	3	4	5	6	7	8	9
Name	Amy	Ben	Cho	Dan	Eva	Fan	Gus	Hal	Ian	Jim
Mark										

To retrieve an element, we would not have to scan any part of the table, but go directly to the record's location by computing its hash value. For example, If Eva wants to know what her mark was, since ascii('E') = 69, we compute (69-5) % 10 which gives 4, the location of Eva's record in the table.

Of course, there are other important considerations, such as the number of elements that can be stored at any given table location (called a bucket), and how to accommodate overflows and handle collisions (two or more elements with the same hash value). It has been pointed out to me by a reviewer of this paper that such considerations are equally important as the derivation of the index. But it can be argued that these are secondary in nature given the motivation of the hashing technique. Its essence can be said to be the derivation of a number from a given key, which is then subsequently used to index into an array where the element is stored with the purpose of eliminating a scan of any part of the array.

3 The Katapayadi Scheme

In classical India, letters of the Sanskrit alphabet were initially used to represent numbers. The grammarian Panini (4th or 5th century BC) who is believed to have written the first generative grammar for a natural language (Asher 1994) assigned the values 1 through 9 and 0 to the Sanskrit vowels a, i, u, etc. For example, *Sutra* (rule) v.i.30 of his grammar, *Ashtadhyayi*, is marked with the letter i, which indicates that the rule applies to the next two rules (Datta and Singh 1962, p.63). It is also known that various synonyms for the number words existed. In one system, words with meanings evocative of the numbers they represented were used. For example, the words indu (moon), dhara (earth) etc. stood for the number one since there was only one of each, netra (eyes), paksha (wings), etc. stood for two and so on. A more comprehensive list of such synonyms can be found in (Ifrah 1985, p.446) who also gives the following instance of its use by Bhaskara I who in 629 A.D. wrote the number 4,320,000 as vijadambaraːkaːɕaɕuːnjajamaraːmaveda or *sky/atmostphere/space/void/primordial couple/Rama/Veda* = 0000234. The term has been transliterated from the Sanskrit using the International Phonetic Alphabet. The palatal sibilant, commonly transcribed as ś is represented using ɕ conforming to the guidelines in (Halle and Clements 1983).

The Katapayadi scheme was initially just another such system of expressing numbers through the use of letters (Sanskrit consonants in this case), with more than one synonym for each number. The consonants themselves were unevocative of the values they represented unlike the earlier scheme, but they now possessed the powerful ability to form easily memorisable words through the insertion of vowels between them. Meaningful and mnemonic words could now be formed using these letters in much the same way as mnemonic words are coined today to represent commercial telephone numbers. In this sense, the Katapayadi scheme could be seen as just a mnemonic technique to help remember numbers, or at best, a coding scheme like ASCII to derive numeric values from non-numeric tokens, but it is noteworthy that the scheme continued to be used long after the invention of numeric symbols and during this time was put to several applications. It is the application of the scheme to the particular instance described in the next section which is remarkably similar to that of modern hashing.

The following Sanskrit verse describes one version of the Katapayadi scheme. (Fleet 1911) quotes this from C.M.Whish (Trans. Lit. Soc. of Madras, Part 1, p.57, 1827) who quotes this from an unspecified source, but (Datta and Singh 1962) state that it is found in *Sadratnamala*, which is a treatise on astronomy published in 1823 by Prince Sankaravarman of Katattanat in North Malabar. The prince was an acquaintance of Mr Whish who spoke of him in high terms as "a very intelligent man and acute mathematician" (Raja 1963). *Sadratnamala*

Value	1	2	3	4	5	6	7	8	9	0
Velar and Palatal Stops	k	k^h	g	g^h	ŋ	c	c^h	ɟ	ɟ^h	ɲ
Retroflex and dental stops	ʈ	ʈ^h	ɖ	ɖ^h	ɳ	t	t^h	d	d^h	n
Labial stops	p	p^h	b	b^h	m					
Fricatives & Glides	j	r	l	v	ɕ	ʂ	s	h		

Table 1: The Katapayadi translation table

was published with a commentary in the Malayalam monthly *Kavanodayam*, vol.16, 1898, Calicut.

> naɲavacaɕca ɕunjani samk^{h}ja kaʈapajaːdajah |
> miɕre tuːpaːnta hal samk^{h}ja na ca cintjo halasvarah ||

(ɲ and n denote zeroes; the letters (in succession) beginning with k, ʈ, p and j (the palatal glide, y in non-phonetic representation) denote the digits. In a conjoint consonant, only the last one denotes a number; and a consonant not joined to a vowel should be disregarded)

There are said to be four variations of this scheme, which is claimed as the reason for its not coming into general use. The transcription scheme is more easily understood from the table 1. It lists the Sanskrit consonants, with their associated numeric values as specified in the verse. Each of the lines except the last consists of stops in the following sequence - unvoiced and unaspirated, unvoiced and aspirated, voiced and unaspirated, voiced and aspirated, and nasal. In the first line the velars are followed by the palatals and in the second line, the retroflexes are followed by the dentals. The last line consists of fricatives.

The following interesting verse also appearing in *Sadratnamala*, illustrates an application of the scheme:

> b^{h}adrambud^{h}isidd^{h}aɟanmagaɳitaɕradd^{h}aːmajadb^{h}uːpagih

If we translate this using the procedure described earlier in the verse about the scheme, we get

> b^h = 4 (from table)
>
> dr = 2 (only the last part of the conjoint consonant, r, is considered)
>
> mb = 3 (similarly, only the b of mb is considered), etc.

This gives the final value 423979853562951413. Since it is known that traditional Indian practice was to write number words in ascending powers of 10 (Ifrah 1985, p.445) (Menninger 1969, pp.398–399), the number represented above, properly, is 314159265358979324 which is recognisable to be just the digits of pi to 17 places (except that the last digit is incorrect—it must be 3).

(Menninger 1969, p.275) also quotes an example[1] of the Indian name for the lunar cycle being *anantapura*, which in addition to having semantic content itself, also gives the Katapayadi value 21600 (using the consonants n-n-t-p-r), which is the number of minutes in the lunar half-month ($15 \times 25 \times 60$).

The originator of this scheme is not known, as with many other Indian inventions and discoveries, but it is believed that the scheme was probably familiar to the Indian mathematician and astronomer Aryabhata I in the 5th century A.D. and to Bhaskara I who lived in the 7th century A.D. (Sen 1971, p.175). The oldest datable text that employs the scheme is *Grahacaranibandhana*, written by Haridatta in 683 AD (Sarma 1972, pp.6–8). The scheme is said to have been used in a wide variety of contexts, including occultisms like numerology. A large number of South Indian chronograms have been composed using this scheme (see for eg. Epigrahia Indica, 3: p.38, 4: pp.203–204, 11: pp.40–41, 34: pp.205–206). It is also said that the Indian philosopher of the 7th Century, Sankara, was named such that the Katapayadi value of his name gives his birthday—215, indicating the fifth day of the first fortnight of the second month in the Indian lunar calendar (Sambamurthy 1983). Not much else is known about the status or application of this scheme since then. But in the 18th century, we find a novel revival of it in South Indian musicology which is arguably similar to modern hashing. This is described in the following section.

4 An Application of the Katapayadi Scheme

In classical South Indian music, the raga is roughly equivalent to the Western chord. These ragas are classified according to a unique scheme. What follows is a brief description of this classification as is pertinent to the subject of this paper. A more comprehensive treatment of Indian musicology, its concepts and terms can be found in (Wade 1969).

A raga can either be a *Janaka* (root) raga or a *Janya* raga which is considered to be a descendant of one of the *Janaka* ragas. The scale of a *Janaka* raga has seven notes in its ascent and the same seven notes in reverse in its descent. A *Janya* raga is a modification of its parent *Janaka* raga through the insertion or deletion of one or more notes and/or possibly the re-ordering of some notes in either or both the ascent and descent of the scale. The seven notes are respectively called Sa (*Shadjam*), Ri (*Rishabham*), Ga (*Gandharam*), Ma (*Madhyamam*), Pa (*Panchamam*), Da (*Dhaivatam*) and Ni (*Nishadam*). These are the equivalents of the western solfa syllables Do, Re, Mi, Fa, So, La and Ti. The notes Sa and Pa (the fifth) are considered fixed, and must occur unchanged in all the *Janaka* ragas. If we consider the octave to consist of the 12 notes C, C#, D, D#, E, F, F#, G, G#, A, A# and B, since C and G are fixed, Ri and Ga can take any combination of two notes from C#, D, D# and

[1] It has been pointed out to me by Dr Takao Hayashi, Science and Engineering Research Institute, Doshisha University, Japan, in a personal communication, that this does not follow from the most popular Katapayadi coding scheme since in the conjoint consonant *nt*, only the *t* should denote a number

E. Similarly Da and Ni can take any combination of two notes from G#, A, A# and B and Ma can take any of the two values F or F#. Thus there can be a total of $2 \times^4 C_2 \times^4 C_2 = 72$ possible *Janaka* Ragas. If we arrange these ragas systematically in a table, it is possible to derive the notes used by any one of them from its index in the table. Accordingly, the table of 72 ragas is constructed as follows: The first 36 ragas in the table use F as the middle note Ma, and the second 36 use F#. In other respects they are identical. Each half of the table is further divided into 6 sections called Chakras, each of which has 6 ragas in it. Each of the 6 Chakras in each half use one of the 6 possible combinations of the notes Ri and Ga, while, within each Chakra, the notes Ri and Ga remain constant, but Da and Ni take on each of their 6 possible combinations. Thus the arrangement in table 2 is arrived at.

This classification makes it easy for us to determine the notes of a raga given its serial number in the table. For example if we were asked to play the scale of raga number 65, we would know that it uses the note F# since 65 div 36+1=2. Since 65 % 36=29 and 29 div 6+1=5, we would know that it uses the fifth possible combination of Ri and Ga which is D & E. Also since 26 mod 6=5, we know it uses the fifth possible combination of Da and Ni which is A and B. Thus the scale of Janaka Raga number 65 is: C, D, E, F#, G, A and B.

This means that given the name of a raga, one need only search for its raga number. The notes can be mechanically derived from its number. However, an Indian raga has certain additional musical properties other than the notes it uses. Frequently, also, a janya raga which inherits some properties from its janaka raga is described in terms of the modifications done to its parent which resulted in that particular raga. These are usually discussed under a description of the Janaka raga and its descendants, or in concise forms, given succinctly alongside its name in a table. To get complete information about a Janaka Raga, then, a table search to find its position given its name is presupposed. Things would be even simpler if we were able to derive the number of a raga directly from its name. This is precisely what was done by the South Indian musicologists. Each raga was named in such a way that a Katapayadi translation of the first two syllables of its name gives us its number in the table. For example, the raga Mechakalyani gives us the number 65 (derived from the first two syllables Me and Cha) and Vanaspati gives 4. Thus it is now possible to go directly to the raga's position in its table from its name without having to do a search.

The exact person who coded the names of the ragas thus seems to be in dispute, but it is fairly certain that such a codification was complete by the end of the 18th Century. (Aiyyangar 1972, p.189) states that although Venkatamakhi lays a claim to this arrangement in 1660, it should really be credited to his grandson Muddu Venkatamakhi in the early 18th century who added it as a supplement to the former's work Chaturdandi Prakasika.

F				F#			
Chakra	Ri,Ga	Da,Ni	Raga	Chakra	Ri,Ga	Da,Ni	Raga
		G#,A	1			G#,A	37
		G#,A#	2			G#,A#	38
1	C#,D	G#,B	3	7	C#,D	G#,B	39
		A,A#	4			A,A#	40
		A,B	5			A,B	41
		A#,B	6			A#,B	42
		G#,A	7			G#,A	43
		G#,A#	8			G#,A#	44
2	C#,D#	G#,B	9	8	C#,D#	G#,B	45
		A,A#	10			A,A#	46
		A,B	11			A,B	47
		A#,B	12			A#,B	48
		G#,A	13			G#,A	49
		G#,A#	14			G#,A#	50
3	C#,E	G#,B	15	9	C#,E	G#,B	51
		A,A#	16			A,A#	52
		A,B	17			A,B	53
		A#,B	18			A#,B	54
		G#,A	19			G#,A	55
		G#,A#	20			G#,A#	56
4	D,D#	G#,B	21	10	D,D#	G#,B	57
		A,A#	22			A,A#	58
		A,B	23			A,B	59
		A#,B	24			A#,B	60
		G#,A	25			G#,A	61
		G#,A#	26			G#,A#	62
5	D,E	G#,B	27	11	D,E	G#,B	63
		A,A#	28			A,A#	64
		A,B	29			A,B	65
		A#,B	30			A#,B	66
		G#,A	31			G#,A	67
		G#,A#	32			G#,A#	68
6	D#,E	G#,B	33	12	D#,E	G#,B	69
		A,A#	34			A,A#	70
		A,B	35			A,B	71
		A#,B	36			A#,B	72

Table 2: Classification of Root *Janaka* Ragas

5 Discussion

From an observation of the Katapayadi scheme, it seems that there are several important differences between it and modern hashing techniques. Notably, a hashing formula gives a valid bucket number for any given name, but the Katapayadi scheme only gives meaningful results for some names. For example, a true hashing algorithm will never give a number greater than 72 in the above application, whatever the value hashed, but the Katapayadi scheme will.

A hashing algorithm can also take any input and return a number corresponding to its position in a table, whereas in the application of the Katapayadi scheme above the names of the ragas have been carefully chosen for the purpose. Thus it seems more probable that the Katapayadi formula was intended as a mnemonic technique to help people remember long numbers. Indeed, the verse from *Sadratnamala* coding the digits of pi seem to imply just that. In this sense, the scheme is an exact opposite of the modern hashing technique which aims to derive numbers from names, since it aims to derive names from numbers.

But then its application in South Indian musicology, where there are only 72 admissible root ragas is clearly directed at liberating the table—lookup operation from the constraints imposed on it by the size of the table. This is the basic aim of a hashing technique. A good hashing algorithm seeks to perform the operations of insertion, deletion and lookup with constant time complexity. The insert and delete operations are irrelevant to the application outlined above since the raga names were deliberately coined and already inserted into the table. But once the table had been constructed, lookup took a constant time because of the application of the Katapayadi scheme. The motivation for this must have been similar as for a situation that warrants the application of a hashing strategy now—constant time table lookup. The result too is the same. Here, it is obvious that it bears a strong similarity to the modern hashing technique. To be sure, the Katapayadi scheme was initially developed as a mnemonic technique given the oral culture of education in early India. Indeed, Sir Monier Williams remarks that even the grammar of Panini was mainly intended to aid the memory of teachers than learners by the briefest possible suggestions (Williams 1969). Nevertheless, it is possible for such a mnemonic technique to gradually evolve into a scheme that bears strong similarity to our modern hashing technique. It is relatively easy for one to look at this particular application of the Katapayadi scheme and to come up with a hashing strategy for some modern requirement. But whether the scheme actually influenced later development of the hashing technique is in doubt. It is not certain whether any Indian scholar with knowledge of this technique was a close associate of any of the proponents of early hashing. Nor is it likely that the proponents of hashing knew about the Katapayadi technique. Thus the most we can say at this stage is that the Katapayadi scheme can be thought of as an early precursor to the modern hash functions and its application in South Indian musicology bears in retrospect an interesting similarity to modern hash tables.

6 Acknowledgments

The author is grateful to Richard Salomon, Department of Asian languages and literature, University of Washington and Takao Hayashi, Science and Engineering Research Institute, Doshisha University, Japan for help in procuring some references. Dr Hayashi also read the paper and offered several useful comments. Dr John Newman, Department of Linguistics and Second Language Teaching, Massey University, offered much help and advice on transliterating Sanskrit using IPA. ...

References

Aiyyangar, R. (1972). *History of South Indian (Carnatic) music.* Poona: Aryabushan Press.

Asher, R. E. (1994). Panini. In *The encyclopaedia of language and linguistics*, pp. 2916–2918. Oxford: Permagon Press.

Datta, B. and A. N. Singh (1962). *History of Hindu Mathematics, Parts I & II.* Bombay: Asia Publishing House.

Dumey, A. I. (1956). Indexing for rapid random-access memory. *Computers and Automation 5*(12), 6–9.

Fleet, J. F. (1911). The Katapayadi system of expressing numbers. *Journal of the Royal Asiatic Society JRAS 1911*, 788–794.

Halle, M. and G. N. Clements (1983). *Problem book in phonology.* Cambridge, Massachusetts: The MIT Press.

Ifrah, G. (1985). *From one to zero.* New York: Viking Penguin Inc. Transl. Bair,L.

Knuth, D. E. (1973). *The art of computer programming, Vol 3: Searching and Sorting.* Reading, MA.: Addison-Wesley.

Menninger, K. (1969). *Number words and number symbols.* Cambridge, MA.: MIT Press. Transl. Broneer,P.

Raja, K. K. (1963). Astronomy and mathematics in kerala (an account of its literature). *Adyar Library Bulletin 27*, 118–67.

Sambamurthy, P. (1983). *South Indian Music, Book III.* Madras: The Indian music publishing house.

Sarma, K. V. (1972). *A history of the Kerala school of Indian astronomy (In perspective).* Visveshvaranand Indological Series 55. Hoshiapur: Visveshvaranand Institute Publication 580.

Sen, S. N. E. (1971). *A concise history of science in India.* New Delhi: Indian National Science Academy.

Severance, D. G. (1974). Identifier search mechanisms: A survey and generalized model. *Computing Surveys 6*(3), 175–194.

Wade, B. C. (1969). *Music in India: The classical traditions*. Prentice-Hall history of music series. New Jersey: Prentice-Hall.

Williams, S. M. (1969). *Indian Wisdom*, Volume XXXVI of *Chowkamba Sanskrit Studies*. Varanasi: Chowkamba Sanskrit Series office.