Why Does Unicode Need Revision for Indic Scripts?

S. P. Mudur
National Centre for Software Technology,
Juhu, Mumbai 400 049, INDIA

Motivation
Digital Information -- A fundamental shift in recording and communication

Unicode has the goal of providing a digital encoding for recording and communicating information in all the world languages. It has been a constant endeavour of humans to devise means and techniques that enable recording and communication of information in an unambiguous manner. Indians have a rich tradition in this. Let us consider, for example, India's ancient oral method used to record information and pass it down successive generations through the Guru-Shishya mechanism. The "Gurus" (teachers) meticulously recited and their "Shishyas" (disciples) had to repeat the information perfectly with identical pronunciation, intonation etc. so as to be unambiguously interpretable by the listener. As a result the Indian alphabet enjoys phonetic richness, variety and subtlety rarely found in any other civilisation – so many different vowel sounds, consonant groups, nasal sounds (four different forms of the 'T' and 'D' sounds, three forms of the 'S' and so on).

The oral form gave way to the graphic form (written, scribed, etched, printed etc.). The graphic form too has evolved over centuries so as to be unambiguously interpreted by the reader and also when converting to the oral form. This largely explains why letter shape (font) design is such a complex design activity. It is not merely the design of single letter forms but of visualising all their combinations and ensuring their unambiguous interpretation by all readers.

Today, while we are in the throes of an information revolution, there is a fundamental shift in this process of recording and communicating information – from the graphic form to the digital form. This shift is not merely in the means and techniques that computer technology provides, but a shift in the primary user of the information itself. Increasingly computers and humans will be equal partners in the use of information encoded in the digital format. In fact, digitally encoded information is first "read" by the computer and then processed by it to provide visual or speech renderings in a form that is easily accessible to humans. But we also know today, that visual or speech rendering (display/print or speech synthesis respectively) are not the only computer processes that will make use of the digital information. As the world-wide information base in the digital format increases and network-centric computer processing dominates, humans will look forward to digital agents doing much of the searching and sifting of this vast information base and providing us with specific information extracts, summaries, recommendations etc. Should this not be easily possible in Indian languages as well?

What all this implies is the need for a coding scheme in which encoded strings result in unique and consistent interpretation by other computer processes, not just those computer processes that we can list today, but also those that will evolve in this digital information era. It would be totally lacking in vision if we merely think of the digital information format as one that must provide the transfer of words in books onto words on the screen.

It is in this context that the present Unicode scheme for Indic scripts has flaws.

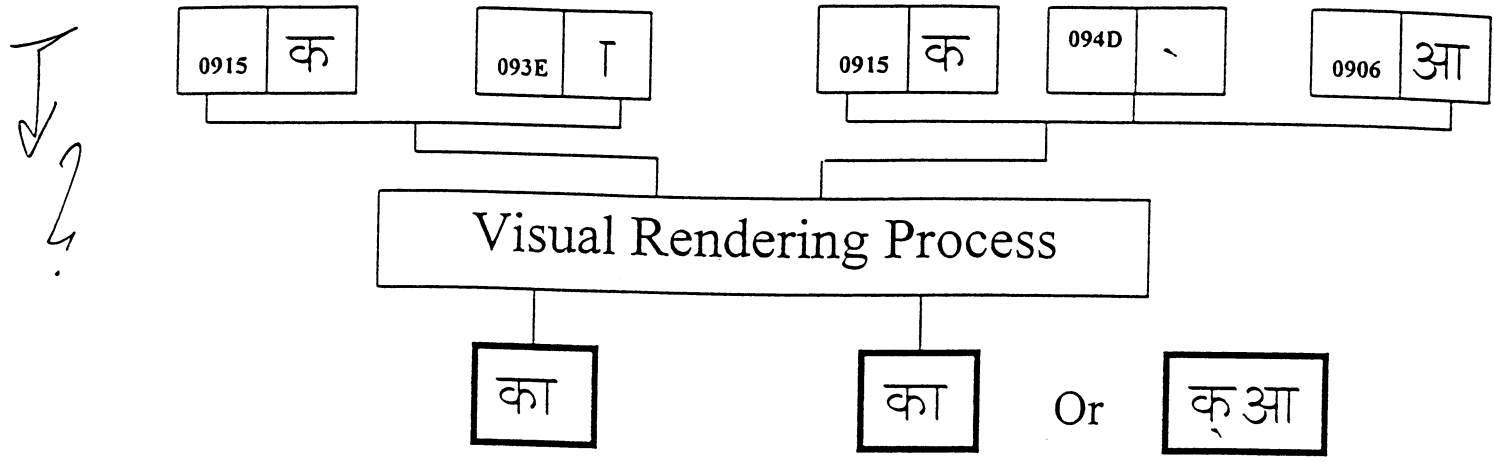
Problems with Unicode for Indic Scripts
Allows ambiguous encoding, inconsistent interpretation and invalid strings

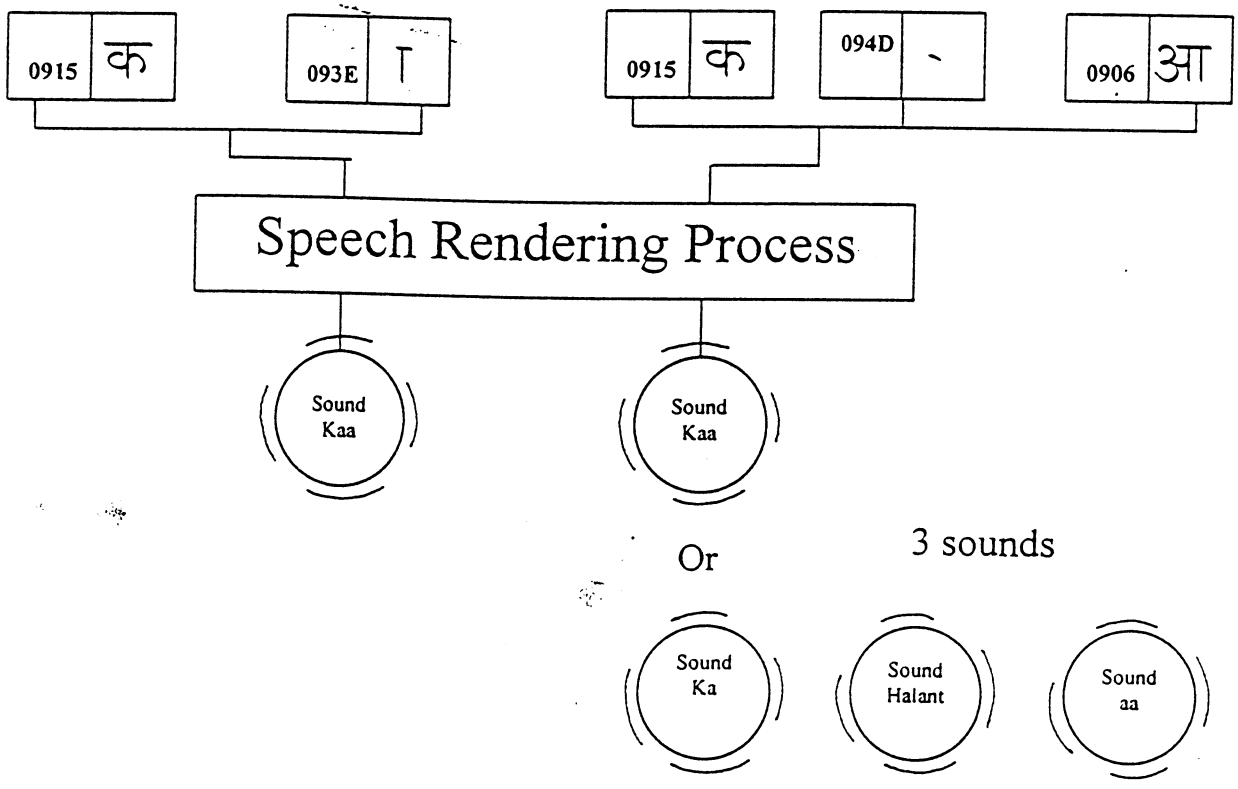
The basic problems arise due to the fact that Unicode for Indic scripts is not minimal and encodes at a level higher than the atomic (metaphorically speaking) level. For brevity, henceforth, reference to Unicode in this document is only for Indic scripts part of Unicode. Three aspects of the present Unicode cause the problems:

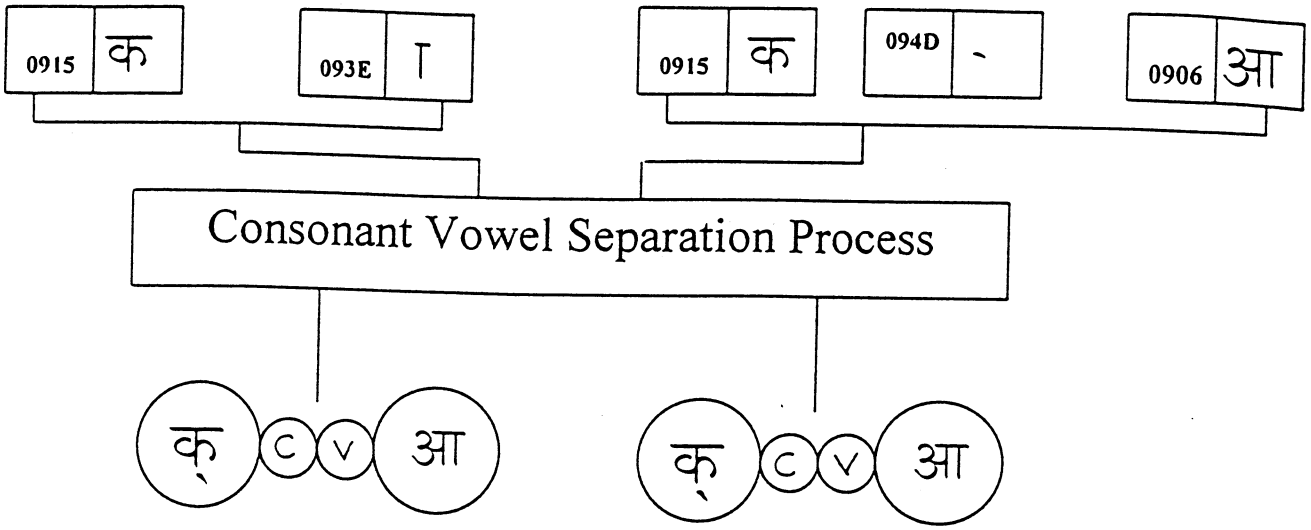
1. Unicode does not encode the pure consonant. Rather the consonant is encoded at a macro level, with a single code denoting a consonant along with the 'A' vowel .
2. It then includes the special character, 'halant' (\ ), to denote the operation of subtracting the 'A' present in the immediately preceding consonant. Here it is important to note that our ancient grammarian Panini has clearly stated that the notional element of 'A' vowel in the consonants is only to facilitate pronunciation of the alphabet. Just as the 'I' vowel sound is used to pronounce consonants like B,C,D,G etc. or the 'Ay' vowel sound in the letters J and K of the English alphabet. The relevant page from an old authoritative book "Siddhanta-kaumudi" is appended. Similarly the use of the 'halant' is meant ONLY for graphically indicating the absence of a vowel in a word-ending, and NOT as the subtractor of the 'A' vowel.
3. The 'matras' are merely different visual renderings of those vowels that immediately follow a sequence of consonants. They are basically vowels, to be pronounced identically. Unicode encodes matras as distinct from vowels. Not only does this result in redundant coding, but more importantly, coupled with the manner in which Unicode encodes consonants, it also necessitates that they be given the special interpretation of replacing the 'A' present in the immediately preceding consonant with the corresponding vowel.

Keeping the above three aspects of Unicode in mind let us consider three computer processes operating on the two Unicode strings shown below. The three computer processes are:

1. The visual rendering process for displaying on screen and printing on paper, film etc.
2. The speech rendering process for digital synthesis of encoded text to speech
3. The consonant-vowel separator process, so essential in all Indian language processing for finding root words, gender, plurality, etc. (like in the Hindi words for boy, girl, boys, "LadkA, Ladki, Ladke", and so on)


0915 क
093E ा
0915 क
094D ्
0906 आ
Visual Rendering Process
का
का
Or
क्आ


0915 क
093E ा
0915 क
094D ्
0906 आ
Speech Rendering Process
Sound Kaa
Sound Kaa
Or
3 sounds
Sound Ka
Sound Halant
Sound aa



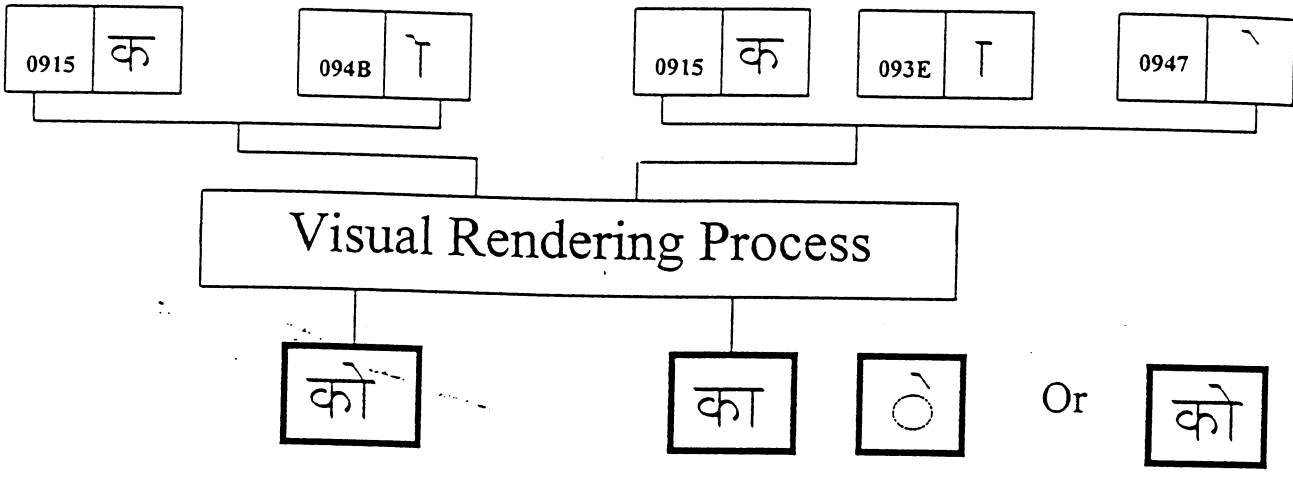
Clearly, the use of the 'halant' with its special interpretation enables the encoding of multiple strings resulting in inconsistent interpretation by the different computer processes. In fact the need to graphically disambiguate the two strings results in an *orthographically inaccurate rendering.* It is a violation of a basic orthographic rule for all Indic scripts, that if a vowel immediately follows a consonant without the embedded vowel then it can ONLY take the form of the matra.

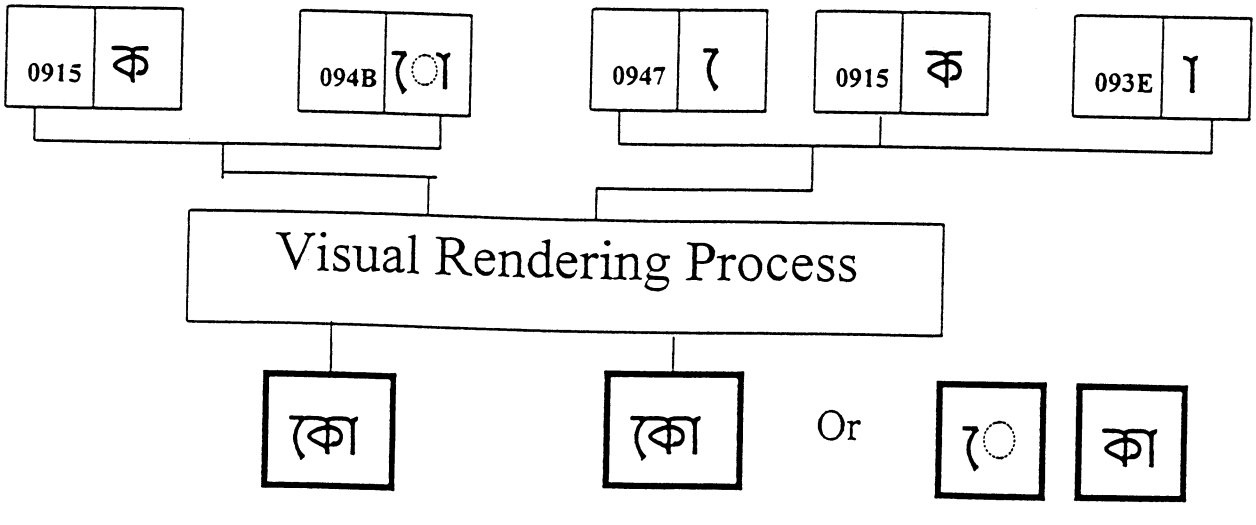
Consider the process of speech input. The sound associated with the 'Aa' matra is the same as the 'Aa' vowel. Which of these two sounds should it produce? Phonetically both strings MUST produce the same sound 'Kaa'. Thus it is inconsistent with the interpretation of the visual rendering process. Unless one forces the *highly counter intuitive* interpretation with the second string as being unpronounceable and 'speaks out' the three Unicode character pronunciations independently.

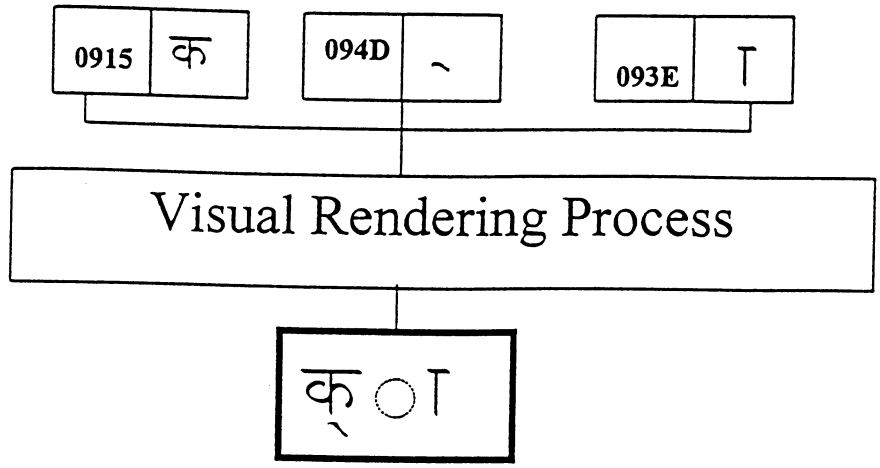
Let's look at the dilemma of the consonant-vowel separator process. If its interpretation has to be consistent with the visual rendering process, then it has to interpret the code 'Ka' followed by 'Aa' matra as the sequence 'Ka', 'halant' followed by the 'Aa' vowel. And *ironically* it would have to interpret the Unicode sequence 'Ka', 'halant' followed by 'Aa' vowel as not the same, which is not possible.

Embedding the 'halant' code with its present interpretation as part of an encoded string is akin to including the 'backspace' character as part of an ASCII string. Imagine the havoc it would cause to all information processing tasks. Moreover, because the 'halant' and the 'matras' are like special context dependent operators with operands, they can be given proper interpretations only if the operands are valid. That is, the 'halant' suppresses the 'A' vowel only if the immediately preceding code is a Unicode consonant code. If it is the vowel 'A' or any other matra or vowel, then it cannot be given any interpretation.

Similarly for any other code that immediately precedes the 'halant', it has no meaningful interpretation. Again, a matra has meaning only if it is immediately preceded by a Unicode consonant. As a result Unicode rendering has to have the concept of invalid code sequences or strings. Consider the two Unicode strings in Devanagari and also in Bengali, and the computer rendering process.







This time we have strings which encode "so called" invalid or illegal character sequences, forcing us to introduce into our scripts the non-phonetic graphic artifact of "invisible consonant" violating the very strong age old phonetic basis of our languages and scripts. Not at all natural. Again what should a speech rendering process do when it encounters such strings ? Ignore them? Treat them as invalid and weed them out of the encoded text ?

Indian language information encoded in the present Unicode code would always have to be preprocessed for disambiguation and validation to enable interpretations consistent with the computer process itself. There is no simple way of guaranteeing consistency in interpretation by different computer processes. Should we carry forward this burden eternally? Particularly, when we have to visualise that computer processing programs can be developed any where in the world, and we would like newly developed applications to be directly applicable to Indian language information as well, it is imperative that one must consider suitably revising Unicode to eliminate such problems.
Particularly, when a robust scheme is inherently present as part of our traditional studies and clearly stated by our ancient grammarians.

The Solution
From our own ancient writings

Panini's classification of the Indian alphabet into pure consonants (*C*) and vowels (*V*), and his simple rendering rules for writing provide us with a beautiful encoding scheme which is devoid of all the above problems. In their simplest form the rendering rules are as follows:

1. In a sequence of vowels and consonants (pure), say *VCCVVCVCCCVVVV*..., the vowel immediately following a consonant must always be rendered in its matra form. All other vowels MUST be rendered in their stand alone form.
2. The sequence of consonants immediately preceding a vowel form a conjunct.

The two rules stated above can be easily built into any shaping engine to give us any

desired quality of display, speech or any other form of computer processing. And we have a highly robust encoding – no ambiguous interpretations, no illegal sequences, no need for the introduction of non-phonetic visual artifacts. Certainly, it was difficult for us Indians to make imported mechanical machinery designed for linear graphic composition to follow these rules. Hence we have had to accept distinct input of vowels, matras and many other graphic forms of the basic alphabet. In the name of technology driven reforms, we Indians have been forced to accept curbing and mutilation of our scripts. But today we have a highly flexible, versatile and programmable rendering process built into the digital computer. This is the period of the digital renaissance for our languages, our scripts, and for recording and communicating of information in our languages. In a multi-lingual country like India with its multiplicity of scripts and languages, some of which may even find official language status with time, it would be truly unfortunate if one resisted or prevented this move to include a robust encoding scheme, one that has been very much part of India's rich tradition for possibly over two thousand years. Considering rendering aspects as external to the basic encoding will also enable us to arrive at a sustainable encoding scheme, that is in sync with current international thinking regarding the encoding of language information.

Today, when India is poised to multiply the use of computers (by a factor of 10 or more in the coming 5 years), when the amount of digital information in Indian languages is expected to increase hundred-fold or even thousand-fold, when the international software manufacturers are coming forward to build in support for Indian languages at the core Operating System level, it would be a technologically retrograde action to resist or prevent the incorporation of a robust encoding scheme that has long term sustainability. My sincere appeal to all computer scientists and IT specialists of India and of the world -- let us build in a system which future generations in India will not find burdensome. Let us pass on a legacy to them. These are extraordinary times and call for extraordinary creative responses.

॥ सारसिद्धान्तकौमुदी ॥

॥ श्रीगणेशाय नमः ॥[1]

नत्वा वरदराजः श्रीपाणिन्यादिमुनित्रयम् ।
करोति[2] बालबोधाय सारसिद्धान्तकौमुदीम् ॥

अ इ उ ण्। ऋ ऌ क्। ए ओ ङ्। ऐ औ च्। ह य व र ट्।
ल ण्। ञ म ङ ण न म्। झ भ ञ्। घ ढ ध ष्। ज ब ग ड द श्।
ख फ छ ठ थ च ट त व्। क प य्। श ष स र्। ह ल्।

इति सूत्राण्यणादिसंज्ञार्थानि । हकारादिष्वकार उच्चारणार्थः ।[3]

१ हलन्त्यम् ॥ १.३.३ ॥
उपदेशेऽन्त्यं हल् इत् स्यात् । उपदेश आद्योच्चारणम् ।

SALUTATION TO LORD GANEŚA

Having made salutation to the triad of sages commencing with Pāṇini, Varadarāja is composing the *Sārasiddhānta-kaumudī* for the enlightenment of the young (scholars).

a i uṇ; etc.

These aphorisms are (meant) for the (formation of) *aṇ* and such other technical designations. The vowel *a* in the letters *ha* etc. (in the aphorisms) is (appended) for (ease of) pronunciation.

1. The final consonant (1.3.3)

In *upadeśa*, the final consonant is mute (*it*). An *upadeśa* (signifies) the original enunciation.

---

१ B_2 श्रीमन्नित्यकुञ्जविहारी राधाकृष्णो विजयतेतमाम् ।

२ D °मि

३ V adds after this लण्मध्ये स्थितसंज्ञकः ।