A Rule Based Syllabification Algorithm for Sinhala

Ruvan Weerasinghe, Asanka Wasala, and Kumudu Gamage

Language Technology Research Laboratory, University of Colombo School of Computing, 35, Reid Avenue, Colombo 7, Sri Lanka
arw@ucsc.cmb.ac.lk, {awasala, kgamage}@webmail.cmb.ac.lk

Abstract. This paper presents a study of Sinhala syllable structure and an algorithm for identifying syllables in Sinhala words. After a thorough study of the Syllable structure and linguistic rules for syllabification of Sinhala words and a survey of the relevant literature, a set of rules was identified and implemented as a simple, easy-to-implement algorithm. The algorithm was tested using 30,000 distinct words obtained from a corpus and compared with the same words manually syllabified. The algorithm performs with 99.95 % accuracy.

1 Introduction

Syllabification algorithms are mainly used in text-to-speech (TTS) systems in producing natural sounding speech, and in speech recognizers in detecting out-of-vocabulary words. The key objectives of this study are to identify the syllable structures in modern Sinhala language and to define an algorithm to syllabify a given Sinhala word to be used in our TTS system. Syllabification algorithms have been proposed for different languages including English, German, Spanish and Hindi, among others. Although a few researchers have documented attempts at syllabifying modern Sinhala words in the Linguistics literature, this is the first known documented *algorithm* for Sinhala syllabification and certainly the first *evaluation* of any syllabification scheme for Sinhala.

Languages differ considerably in the syllable structures that they permit. For most languages, syllabification can be achieved by writing a set of declarative grammatical rules which explain the location of syllable boundaries of words step-by-step. It has been identified that most of these rules adhere to well known theories such as the *Maximum Onset Principle* and the *Sonority Profile*. The association of consonants with the syllable nucleus is derived by the Maximum Onset Principle (MOP).

Maximum Onset Principal: First make the onset as long as it legitimately can be; then form a legitimate coda [2].

Sonority Profile: The sonority of a syllable increases from the beginning of the syllable onward, and decreases from the beginning of the peak onwards [2].

Sonority is related to the acoustic intensity of a sound. Thus, by measuring the acoustic intensities of sounds, the sonority of a sound can be estimated [1]. The classes of vowel and consonant sounds (segments) that are usually distinguished along this dimension are listed in the order of increasing sonority, and this list is referred to as the Sonority Scale [2].

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 438 – 449, 2005.

Sonority Scale:
Obstruents – Nasals – Liquids ([l, r] etc.) – Glides ([w, j]) etc.) – Vowels

Many syllabification algorithms have been developed based on these two theories. For example, the Festival Speech Synthesis System (the framework we use in our Sinhala TTS) by default syllabifies words by finding the minimum sonorant position between vowels [3]. Another sonority scale based syllabification algorithm is presented in detail in [4]. The sonority theory of the syllable does not, however, account for all the phenomena observed in language. Many examples have been provided in the literature to demonstrate this [1], [2]. To avoid the difficulties encountered when using the sonority profile, most of the language specific syllabication schemes are modeled by using finite state machines or neural networks. A multilingual syllabification algorithm using weighted finite-state transducers has been proposed in [5].

In this research, an algorithm to divide Sinhala words into syllables is proposed. The algorithm was tested by using a text corpus containing representative words for each grammatical rule, and its performance was then measured in terms of the percentage of correctly syllabified words. The rest of this paper is organized as follows: Section 2 gives an overview of the Sinhala Phonemic Inventory and Section 3 briefly reviews the linguistic background of modern Sinhala word syllabification including issues we identified and our proposed solutions. Section 4 describes the implementation of the algorithm. The paper concludes with the results & discussion in Section 5.

2 The Sinhala Phonemic Inventory

Sinhala is one of the official languages of Sri Lanka and the mother tongue of the majority (about 74%) of its population. Spoken Sinhala contains 40 segmental phonemes; 14 vowels and 26 consonants as classified below in Table 1 and Table 2 [6].

There are two nasalized vowels occurring in two or three words in Sinhala. They are /ã/, /ã:/, /æ̃/, and /æ̃:/ [6]. Spoken Sinhala also contains the following Diphthongs, /iu/, /eu/, /æu/, /ou/, /au/, /ui/, /ei/, /æi/, /oi/, and /ai/ [7].

A separate letter for vowel /ə/ has not been provided by the Sinhala writing system. In terms of distribution, the vowel /ə/ does not occur at the beginning of a syllable except in the conjugational variants of verbs formed from the verbal stem /kərə/ *(to do.)*. In contrast to this, though the letter "ය්" exists in Sinhala writing system (corresponding to the consonant sound /j/), it is not considered a phoneme in Sinhala.

Table 1. Spoken Sinhala vowel classification

	Front		Central		Back	
	Short	Long	Short	Long	Short	Long
High	i	i:			u	u:
Mid	e	e:	ə	ə:	o	o:
Low	æ	æ:	a	a:		

Table 2. Spoken Sinhala consonant classification

		Labial	Dental	Alveolar	Retroflex	Palatal	Velar	Glottal
Stops	Voiceless	p	t		ʈ		k	
	Voiced	b	d		ɖ		g	
Affricates	Voiceless					c		
	Voiced					j		
Pre-nasalized voiced stops		ᵬ	ᵭ		ᶑ		ᶢ	
Nasals		m		n		ɲ	ŋ	
Trill				r				
Lateral				l				
Spirants		f	s			ś		h
Semivowels		v				y		

3 Syllable Structure

3.1 Methodology

The methodology adopted in this study was to first examine the Sinhala syllable structure from the Linguistics literature to gather the views of scholars from the various linguistic traditions. It was expected that this study would reveal the main issues related to the syllabification of Sinhala and how these issues are addressed by scholars in the literature. This was then subjected to the scrutiny of an algorithm in order to select from among alternative theories.

3.2 Sinhala Syllable Structure

3.2.1 Background

Words in the Sinhala language can be divided in to three groups namely *Nishpanna*, *Thadbhava* and *Thathsama* as described below:

Nishpanna: Words that are of local origin.
Thathsama: Words borrowed from other languages *in their (near) original form.*
Thadbhava: Words derived from other languages but modified to be incorporated to Sinhala (mainly from Sanskrit and Pali).

The high impact of Sanskrit in Sinhala vocabulary formation is due to the fact that, Sinhala and Sanskrit belong to the same Indo-Aryan language family. Further, as the vehicle of Buddhism to Sri Lanka, the Pali (spoken) language has also significantly influenced the vocabulary of Sinhala. Due to various cultural, historical and socio-linguistic factors, other languages such as Tamil, Portuguese, Dutch and English have also impacted the structure and vocabulary of Sinhala.

It is important to note that the *Thathsama*, and *Thadbhava*, categories of words are available in modern Sinhala and are indistinguishable to the layman from words in the *Nishpanna* Category. While no documented evidence exists, it is thought that the

percentage of words in the *Nispanna* category is less than 5%, while the percentage of words in the *Thathsama* and *Thadbhava* categories in Sinhala (the remaining > 95%), are more or less equal. However, for words in the *Thathsama* and *Thadbhava* categories, no official syllable structures were found in literature. This puts the onus on any TTS researcher dealing with Sinhala syllabification to pay urgent attention to the study of the *Thathsama* and *Thadbhava* categories of words.

3.2.2 Syllabification of Words Belonging to the *Nishpanna* Category

It has been identified that there are four legal syllable structures in Sinhala, namely V, VC, CV and CVC for words which belong to the *Nishpanna* category [7]. This can also be represented using the expression: (C)V(C). Though a large number of examples for syllabified words belonging to each of the above structures are presented in the literature [7], the methodology or grammatical rules describing how to syllabify a given word has not been presented. A word can be syllabified in many ways retaining the permitted structures, but only a single correct combination of structures is accepted in a properly syllabified word.

For example, a word having the consonant-vowel structure VCVCVC can be syllabified in the following different ways, retaining the valid syllable structures described in the literature: (V)(CVC)(VC), (VC)(VC)(VC), (VC)(V)(CVC). However, only one of these forms represents the properly syllabified word.

The determination of a proper mechanism leading to the identification of the correct combination and sequence of syllable structures in syllabifying a given word became the major challenge in this research.

Further review of the literature and empirical observation led to the following model with regard to Sinhala syllabification;

1. A fundamental assumption that the accurately syllabified form of a word can be uniquely obtained by formulating a set of rules.
2. That the following set of rules can be empirically shown to be effective.

Syllabification Procedure for the Nishpanna Category

a. Reach the first vowel of the given word and then,
   1. If the phoneme next to this vowel is a consonant followed by another vowel, then mark the syllable boundary just after the first vowel. (Rule #1)
      i.e. a word having a consonant-vowel structure xVCV… Should be syllabified as (xV)(CV…), where x denotes either a single consonant or zero consonant.
   2. If the phoneme next to this vowel is a consonant followed by another consonant and a vowel, then mark the syllable boundary just after the first consonant. (Rule #2)
      i.e. a word having a consonant-vowel structure xVCCV… should be syllabified as (xVC)(CV…), where x denotes either a single consonant or zero consonant.
   3. If the phoneme next to this vowel is another vowel, then mark the syllable boundary just after the first vowel. (Rule #3)
      i.e. a word having a consonant-vowel structure xVV… should be syllabified as (xV)(V…), where x denotes either a single consonant or zero consonant. Only a few words were found in Sinhala having two consecutive vowels in a single word. e.g. “giriulltə” *(Name of city.)*, “aa:və” *(Alphabet)*. The syllable structure (V) mostly occurs at the beginning of the word, except for this type of rare word.

b. Having marked the first syllable boundary, continue the same procedure for the rest of the phonemes as in the case of a new word.
i.e. Repeat the step (a) for the rest of the word, until the whole word is syllabified.

The accuracy of this model was first tested by calculating the syllable boundaries using the examples given in the literature. Convinced that the results were consistent with the *descriptive treatment* of the subject in the literature, it was concluded that the above set of rules could describe an accurate *syllabification algorithm* for words belonging to the *Nishpanna* category.

3.2.3 Syllabification of Words of the *Thathsama* and *Thadbhava* Categories

In the syllabification of foreign words, it has been observed that words borrowed from Sanskrit play a major role compared to those borrowed from other languages. The reason behind this is the presence of a large number of Sanskrit words in Sinhala (about 75% of the *Thathsama* category [8]), and the complexity of codas and onsets of these words when they intermix with the Sinhala phonetic inventory. Defining proper syllabic structures for words borrowed or derived from Sanskrit therefore became a primary focus in this study due to this reason. For this purpose, a carefully chosen list of words in this category (see Appendix A) was presented to recognized scholars of Sinhala and Sanskrit for syllabification and recommendations. A careful analysis of the information and views gathered, led to the identification of a new set of rules distinct from those defined in previous section, on how to syllabify these *borrowed* Sanskrit words. It is proposed that Syllabic structures for words originating from Sanskrit can be represented using the consonant-vowel pattern (C)(C)(C)V(C)(C)(C). It is also noteworthy to mention that the syllabic structures for words belong to the category of *Nishpanna* i.e. (C)V(C) is in fact a subset of this structure.

Languages are unique in syllable structures. Syllabification of words belonging to the categories of *Thathsama* and *Thadbhava* do not completely adhere to the syllabification rules imposed in the language from which the word originated. Syllabification of such words will naturally be altered according to the phonetic inventory and existing syllable structures of the host language, Sinhala in this case. This view was expressed by all of the scholars whom we consulted regarding the syllabification of foreign words. In support of this observation, it was evident that the syllabification of almost all the words borrowed from languages other than Sanskrit (e.g. Pali, Tamil and English) are also consistent with the above set of syllabification rules and syllabic structures produced by them. An algorithm to capture this feature of the language is presented below:

Syllabification Procedure for Thathsama and Thadbhava Category
a. Reach the first vowel of the given word and then,
1. If the vowel is preceded by a consonant cluster of 3, followed by a vowel,
   - If the consonant preceded by the last vowel is /r/ or /y/ then mark the syllable boundary after the first consonant of the consonant cluster. (Rule #4)
     i.e. a word having a consonant-vowel structure xVCC[/r/ or /y/]V.., should be syllabified as (xVC)(C[/r/ or /y/]V…), where x denotes zero or any number of consonants.

- In the above rule, if the consonant preceded by the last vowel is a phoneme other than /r/ or /y/ then,
  - If the first two consonants in the consonant cluster are both stop consonants, then mark the syllable boundary after the first consonant of the consonant cluster. (Rule #5)

  i.e. a word having a consonant-vowel structure xV[C-Stop][C-Stop]CV..., should be syllabified as (xVC)(CCV...), where x denotes zero or any number of consonants.
  - In other situations, mark the syllable boundary after the second consonant of the consonant cluster. (Rule #6)

  i.e. a word having a consonant-vowel structure xVCCCV…, should be syllabified as (xVCC)(CV...), where x denotes zero or any number of consonants.

2. If this vowel is preceded by a consonant cluster of more than 3 consonants ,
   - If the consonant just before the last vowel is /r/ or /y/ then, mark the syllable boundary before 2 consonants from the last vowel. (Rule #7)

     i.e. a word having a consonant-vowel structure xVCCC…[/r/ or /y/]V... should be syllabified as (xVCCC…)(C[/r/ or /y/]V..), where x denotes zero or any number of consonants.
   - In other situations, mark the syllable boundary just after the minimum sonorant consonant phoneme of the consonant cluster. (Rule #8)

     i.e. a word having a consonant-vowel structure xVCCC…CV..., should be syllabified just next to the minimum sonorant consonant in the consonant cluster.

b. Having marked the first syllable boundary, continue the same procedure for the rest of the phonemes as in a new word.

   i.e. Repeat the step (a) for the rest of the word, until the whole word is syllabified.

The words with consonant clusters of more than 3 consonants are rarely found in Sinhala, and to avoid the confusion of syllabification of words in such situations, the algorithm makes use of the universal sonority hierarchy in deciding the proper position to mark the syllable boundary. In these situations, the syllable boundary is marked next to the first occurrence of a minimum sonorant consonant.

3.2.4 Ambisyllabic Words in Sinhala

Some ambisyllabic words are also found in Sinhala. This situation arises due to the fact that some words with complex coda or onset can be syllabified in several ways.

For example, a word such as /sampre:kʃənə/ (transmit) the /p/ can be interpreted as a coda with respect to the preceding vowel, as in /samp/re:k/ʃə/nə/or as an onset with respect to the following vowel, as in /sam/pre:k/ʃə/nə/.

More examples for this kind of word include, /mats/yə/, /mat/syə/ *(fish);* /sank/ya:/, /san/kya:/ *(number)* and /lakʃ/yə/,/lak/ʃyə/*(point).*

Some Sanskrit loan words in Sinhala (including word internal clusters ending in /r/ and preceding a vowel) can either be syllabified by reduplicating the first consonant sound of the cluster or by retaining the original word as in the following examples.

/krə/mak/rə/mə/yə/→ /krə/mak/krə/mə/yə/ *(gradually)*
/ap/rə/ma:/nə/→ /ap/prə/ma:/nə/ *(unlimited)*
/ja/yag/ra:/hi:/→ /ja/yag/gra:/hi:/ *(victory)*

This description demonstrates that the rules and procedures determined above are complied with even by ambisyllabic words. It is important to note that while both forms are acceptable, *one* of these will be provided by the algorithm stated above.

4 The Syllabification Algorithm

The rules identified in sections 3.2.2 and 3.2.3 are sensitive to the sequence since they interact with each other. In this section, the Sinhala syllabification rules identified above are presented in the form of a formal *algorithm*. The function *syllabify()* accepts an array of phonemes generated by our *Letter-To-Sound* module[1] for a particular word, along with a variable called *current_position* which is used to determine the position of the given array currently being processed by the algorithm.

Initially the *current_position* variable will be initialized to 0. The *syllabify()* function is called recursively until all phonemes in the array are processed. The function *mark_syllable_boundary*(*postion)* will mark the syllable boundaries of an accepted array of phonemes. The other functions used within the *syllabify()* function are described below.

- *no_of_vowels(phonemes):* accepts an array of phonemes and returns the number of vowels contained in that array.
- *is_a_vowel(phoneme):* accepts a phoneme and returns true if the given phoneme is a vowel.
- *count_no_of_consonants_upto_next_vowel(phonemes, position):* accepts an array of phonemes and a starting position; and returns the count of consonants from the starting position of the given array until the next vowel is found.
- *is_a_stop(phoneme):* returns true if the accepted phoneme is a stop consonant.
- *find_min_sonority_position(phonemes, position):* returns the minimum sonorant position of a given array of phonemes, by starting the search from the given position.

A complete listing of the algorithm is provided in Appendix B.

5 Results and Discussion

Our algorithm was tested on 30,000 distinct words extracted from the (unbalanced) *UCSC Sinhala Corpus BETA*, and compared with correctly hand syllabified words. Text obtained from the category "*News Paper > Feature Auricles > Other*" was chosen for testing the algorithm due to the heterogeneous nature of these texts and hence the perceived better representation of the language in this section of the corpus[2]. A list

[1] Discussed in another paper in preparation.
[2] This accounts for almost two-thirds of the size of this version of the corpus.

of distinct words was first extracted, and the 30,000 most frequently occurring words chosen for testing our algorithm.

The 30,000 words yielded some 78,775 syllables which were distributed as follows among the 8 rules of the algorithm given: Rule #1: 67,350; Rule #2: 10,899; Rule #3: 71; Rule #4: 324; Rule #5: 28; Rule #6: 77; Rule #7: 21 and Rule #*: 5. Note however that owing to the syllable structure of words in the *Nishpanna* category being a subset of those of the *Thathsama* and *Thadbhava* categories, nothing can be inferred about the actual percentages of words in each category from this analysis alone.

The algorithm achieves an overall accuracy of 99.95% when compared with the same words manually syllabified by an expert. An error analysis revealed the following two types of errors:

1. Words composed by joining two or more words (i.e. Single words formed by combining 2 or more distinct words; such as in the case of the English word "*thereafter*"). In this case, syllabification needs to be carried out separately for each word of the compound word, and then concatenated to form a single syllabified word.
2. Foreign (mainly English) words directly encoded in Sinhala.

A detailed study of Sinhala syllabification is presented in the research above. Though a great number of diverse algorithms have been proposed for syllabification in different languages, to the best of our knowledge this is the first such study for Sinhala syllabification proposing a formal algorithm describing the process. The initial study of the literature revealed certain unresolved issues which this study resolved with the aid of scholars. A significant task was carried out in identifying valid syllable structures for words borrowed from Sanskrit. A major effort was also made in identifying and defining a formal set of linguistic rules for syllabification, and then translating same into a simple and easy-to-implement algorithm. Finally, the effectiveness of the proposed algorithm was demonstrated using a set of words extracted from a Sinhala corpus.

Syllabification is an important component of many speech and language processing systems, and this algorithm is expected to be a significant contribution to the field, and especially to researchers working on various aspects of the Sinhala language.

Acknowledgement

This work has been supported through the PAN Localization Project, (*http://www.PANL10n.net*) grant from the International Development Research Center (IDRC), Ottawa, Canada, administered through the Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences, Pakistan. The authors are indebted to Sinhala Language scholars, Prof. Wimal G. Balagalle, Prof. S.A.G. Wijesinghe, Prof. R.M.W. Rajapaksha, and Prof. J.B. Dissanayake for their invaluable support and advice throughout the study. We also wish to acknowledge the contribution of Mr. Viraj Welgama, Mr. Dulip Herath, and Mr. Nishantha Medagoda of Language Technology Research Laboratory of the University of Colombo School of Computing, Sri Lanka.

References

1. Ladeforged, P., A Course In Phonetics, 3rd edn., Harcourt Brace Jovanovich College Publishers, 301, Commerce Street, Suite 3700, Fort Worth TX 76102 (1993)
2. Gussenhoven, C., Jacobs, H., Understanding Phonology, Oxford University Press Inc, 198, Madison Avenue, New York, NY 10016 (1998)
3. Black, A.W., Taylor, P., Caley R., The Festival Speech Synthesis System: System Documentation, University of Edinburgh, Edinburgh (1999)
4. Brasington, R., "A simple syllabifier", (LOGO and natural language), Available: http://www.personal.rdg.ac.uk/~llsling1/Logo.WWW/Sound.patterns/Simple.syllabifier.html,(Accessed: 2005, February, 02)
5. Kiraz, G.A., Mobius, B., "Multilingual Syllabification Using Weighted Finite-State Transducers", Bell Labs – Lucent Technologies, Murray Hill, NJ 07974, USA, In Proceedings of the Third ESCA Workshop on Speech Synthesis (Jenolan Caves, Australia, 1998) (1998)
6. Karunatillake, W.S., An introduction to spoken Sinhala, 3rd edn., M.D. Gunasena & Co. ltd., 217, Olcott Mawatha, Colombo 11 (2004)
7. Disanayaka, J.B., The structure of spoken Sinhala, National Institute of Education, Maharagama (1991)
8. Jayathilake, K., *Nuthana Sinhala Vyakaranaye Mul Potha*, Pradeepa Publications, 34/34, Lawyers' Office Complex, Colombo 12, (1991)

Appendix A: Word-List

කූටෝපක්‍රම	k u: t o: p a k k r ə m ə
ප්‍රතිඝජන	p r u t a g j a n ə
ක්‍රමක්‍රමයෙන්	k r ə m a k k r ə m ə y e n
ස්ත්‍රීන්	s t r i: n
විද්‍යාඥයා	v i d y a: j ɲ ə y a:
කෘමියා	k r u m i y a:
පිතෘන්	p i: t t r u: n
සෞභාග්‍ය	s a u b a: g y ə
කාව්‍යෝපදේශය	k a: v y o: p ə d e: ʃ ə y ə
අවිද්‍යාව	a v i d y a: v ə
ප්‍රඥාව	p r a j ɲ a: v ə
කොන්ස්තන්තිනෝපලය	k o ŋ s t a n t i n o: p ə l ə y ə
ස්වප්න	s v a p n ə
ශල්‍යකර්ම	ʃ a l y ə k a r m ə
පාර්ලිමේන්තුව	p a: r l i m e: n t u w ə
ද්වන්ද්ව	d v a n d v ə
හෘදස්පන්දනය	h ə r d ə s p a n d ə n ə y ə
සම්ප්‍රේක්ෂණ	s a m p r e: k ʃ ə n ə
ක්ෂේත්‍ර	ʃ e: ʃ t r ə
ප්‍රවෘත්ති	p r ə v u r t i
ප්‍රව්‍රජ්‍යා	p r ə v u r j y a:

ප්‍රඥාඩිය	p r ə ʃ r a b d i y ə
සංස්කෘත	s a n s k r u t ə
springs	s p r iŋ g s
scratched	s k r æ c ɖ
straights	s t r e: i t s
strength	s t r e n t s
postscript	p o: s t s k r i p t
area	e: r i a:

Appendix B: Syllabification Algorithm

```
function syllabify (phonemes, current_position)
 if no_of_vowels(phonemes) is 1 then
  mark_syllable_boundary(at_the_end_of_phonemes)
 else
  if is_a_vowel(phonemes[current_position]) is true
  then
   no_of_consonants=
   count_no_of_consonants_upto_next_vowel
   (phonemes,current_position)
   if no_of_consonants is 0 then
    if is_a_vowel(phonemes[current_position+1]) is
    true then
     mark_syllable_boundary(current_position) % Rule#3
     syllabify(phonemes, current_position+1)
    end if
   else
    if no_of_consonants is 1 then
     mark_syllable_boundary(current_position) % Rule#1
     syllabify(phonemes, current_position+2)
    end if
    if no_of_consonants are 2 then
     mark_syllable_boundary(current_position+1)
     syllabify(phonemes, current_position+3) % Rule#2
    end if
    if no_of_consonants are 3 then
```

```
if phonemes[current_position+3] is
( "r" or "y") then
  mark_syllable_boundary(current_position+1)
  % Rule#4
  syllabify(phonemes, current_position+4)
else
  if is_a_stop(phonemes[current_posi+1]) is
  true and is_a_stop(phonemes[current_posi+2])) is
  true then
      mark_syllable_boundary(current_position+1)
     % Rule#5
      syllabify(phonemes, current_position+4)
     else
      mark_syllable_boundary(current_position+2)
      % Rule#6
       syllabify(phonemes, current_position+4)
       end if
     end if
    end if
  if no_of_consonants are greater than 3 then
   if phonemes[current_position+no_of_consonants]
   is( "r" or "y") then
     mark_syllable_boundary
     (current_position+no_of_consonants-2) % Rule#7
     Syllabify
     (phonemes, current_position
     +no_of_consonants-1)       else
    syllable_boundary=find_min_sonority_position
                     (phonemes,current_postion)
     mark_syllable_boundary(syllable_boundary)
     % Rule#8
     Syllabify
     (phonemes, syllable_boundary+1)
   end if
  end if
```

```
     end if
    else
     temp=current_postion
     repeat
      temp = temp + 1;
     until(is_a_vowel(phonemes[temp]) is true
     syllabify(phonemes,temp)
    end if
end if
```