Computational modelling of Yorùbá numerals in a number-to-text conversion system

Olúgbénga O. Akinadé and Ọdẹ́túnjí A. Ọdéjọ́bí
Computing and Intelligent Systems Research Group
Department of Computer Science and Engineering
Ọbáfẹ́mi Awólọ́wọ̀ University
Ilé-Ifẹ̀, Nigeria

ABSTRACT

In this paper, we examine the processes underlying the Yorùbá numeral system and describe a computational system that is capable of converting cardinal numbers to their equivalent Standard Yorùbá number names. First, we studied the mathematical and linguistic basis of the Yorùbá numeral system so as to formalise its arithmetic and syntactic procedures. Next, the process involved in formulating a Context-Free Grammar (CFG) to capture the structure of the Yorùbá numeral system was highlighted. Thereafter, the model was reduced into a set of computer programs to implement the numerical to lexical conversion process. System evaluation was done by ranking the output from the software and comparing the output with the representations given by a group of Yorùbá native speakers. The result showed that the system gave correct representation for numbers and produced a recall of 100% with respect to the collected corpus. Our future study is focused on developing a text normalisation system that will produce number names for other numerical expressions such as ordinal numbers, date, time, money, ratio, etc. in Yorùbá text.

Keywords:
Analysis of numerals, Yorùbá numerals, numbers to text, text normalisation

Journal of Language Modelling Vol 2, No 1 (2014), pp. 167–211

1 INTRODUCTION

The use of numbers and their power in capturing concepts makes them indispensable in effective communication (Goyvaerts 1980). In any society, the use of numbers is firmly anchored to numerous beliefs and perceived usefulness of the significant philosophy underlying numerical messages (Abímbọ́lá 1977). In fact, key advancement in civilisation can be traced to the conception, invention, representation, and manipulation of numbers to facilitate accurate rendering of measurable objects. This has made the use of numbers an important tool within the society, where it is used in trade, cosmology, mathematics, divination, music, medicine, etc. Early cultures devised various means of number representation, which include body/finger counting (Zaslavsky 1973; Saxe 1981), object counting, Egyptian numerals, Babylonian numerals, Greek numerals, Chinese numerals, Roman numerals, Mayan numerals, Hindu-Arabic numerals, etc. The Hindu-Arabic numeral system, which is considered to be the greatest mathematical discovery (Bailey and Borwein 2011), is still the most commonly used symbolic representation of numbers due to its simplicity and the fact that it requires little memorisation to represent practically any number.

Naming numbers in human languages requires various mathematical and linguistic processes. For example, the number 74 is represented as 70 (7×10) increased by 4 in English, whereas it is represented as 60 (6×10) increased by 14 ($4 + 10$) in French. In Logo, the number 74 is represented as 10 added to 60 (20×3) increased by 4. In Yorùbá, in turn, the same number is derived in a more complex way by adding 4 to 80 (20×4) reduced by 10. Table 1 shows the representation of the number 74 in the four languages.

The analysis of number names is important but understudied in human language processing. While it may seem trivial to compute number names in languages like English, it may be difficult to get it

Table 1: Derivation of the number 74 in four languages

Language	**Name**	**Derivation**
English	*seventy four*	$(7 \times 10) + 4$
French	*soixante-quatorze*	$60 + 14$
Logo	*nyaa̱ na drya mudri drya su*	$(20 \times 3) + 10 + 4$
Yorùbá	*ẹ̀rìnlélàádọ́rin*	$4 + (-10 + (20 \times 4))$

right in many other languages, particularly in the Yorùbá language. In this paper, we present a formal description of the Yorùbá numeral system; specifically, the problem of Yorùbá number name transcription is addressed from an engineering perspective, by applying standard theories and techniques to an understudied language. This is part of a wider interest in the development of Text-To-Speech (TTS) synthesis and Machine Translation (MT) systems for the Yorùbá language. In TTS and related applications, text normalisation is often the first task, in which Non-Standard Words (NSW) such as numbers, abbreviations, acronyms, time, date, etc. are correctly identified and expanded into their textual forms (Sproat 1996). The expansion of numerical expressions in text is thus a key task in such applications because numbers occur more frequently in varying forms within a block of text. These forms include cardinal numbers, ordinal numbers, telephone numbers, date, time, percentages, monetary value, address, etc.

The rest of this paper is structured as follows: Section 2 gives an analysis of the Yorùbá numeral system and its associated number naming rules. Section 3 discusses the system design and implementation, while Section 4 discusses the results. Section 5 presents the system evaluation and Section 6 concludes the paper with areas of further study.

2 THE YORÙBÁ NUMERALS

The Yorùbá language (ISO 639.3 yor), which belongs to the West Benue-Congo branch of the Niger-Congo African languages family, is spoken by about 19,000,000 speakers in the South-Western Nigeria (Owólabí 2006). The language is also spoken in other West African countries such as Central Togo, the East-Central part of the Republic of Benin, and Creole population of Sierra Leone. Outside Africa, Yorùbá (called Nagô, Aku, or Lukumi; Lovejoy and Trotman 2003) is spoken in Brazil, Cuba, and Trinidad and Tobago.

Without a formal method of documenting literature, the Yorùbá community developed a complex numeral system that extensively uses subtraction throughout its system (Verran 2001). This has attracted many linguistic scholars to investigate the reasons why this community has developed an intricate numeral system. Certainly, knowledge of the Yorùbá numeral system has been passed from generation to

generation by means of oral literature. Young language learners, in particular, are made to undergo drills of reciting rhymes with numbers ranging from 1 to 10.

In an early study of the Yorùbá numeral system, Mann (1887) shows how large numbers could be represented as an arithmetic combination of the basic number units and reveals that the subtraction operation plays an important role in number naming. The peculiarity in the Yorùbá numerals was highlighted as follows:

> *"Very different is the framework of the Yorùbá, it can boast of a greater number of radical names of numerals, and to a large extent makes use of subtraction..." (Mann 1887, p. 60)*

A fact worth noting is that some systems illustrate a pervasive use of the subtractive techniques. Examples of such systems are the clock system and the Roman numeral system. In the conventional clock system, when the minute part of time is greater than 30 minutes, the spoken representation can be derived by employing the subtractive technique. For instance, four canonical representations of 2:30 PM are:

(i) Half two (half hour past two)
(ii) Two thirty (2 o'clock + 30 min)
(iii) Thirty minutes after two (2 o'clock + 30 min)
(iv) Thirty minutes to three (3 o'clock − 30 min)

All four representations in (i) to (iv) are acceptable and none has precedence over the other. The form in (iv) is used to a large extent in our daily lives without any difficulty. Similarly, *halb zwei* in German means 'half of the second hour', which is 'half one'. So, the Yorùbá's use of subtraction is not completely exceptional, but its extensive usage may seem unusual, especially when it is preferred over the simpler addition operation.

Another observable feature of the Yorùbá numeral system is the use of base 20 (vigesimal), which likely stems from the counting of cowry shells as described by Mann:

> *"Here we may explain the origin of this somewhat cumbersome system; it springs from the way in which the large sum of money (cowries) are counted. When a bagful is cast on the floor, the*

> *counting person sits or kneels down beside it, takes 5 and 5 cowries and counts silently, 1, 2, up to 20, thus 100 are counted off, this is repeated to get a second 100, these little heaps each of 100 cowries are united, and a next 200 is, when counted, swept together with the first" (Mann 1887, p. 63)*

However, there are vigesimal systems that do not have any relation to cowry shells. A more obvious reason for vigesimal systems could be that humans have 10 fingers and 10 toes. The use of 20 as a base may seem cumbersome, however, it is not entirely exceptional. In many languages, especially in Europe and Africa, 20 is a base with respect to the linguistic structure of the names of certain numbers. Even so, a consistent vigesimal system based on the powers of 20, i.e.: 20, 400, 8000, etc. is not generally used. Examples of a strict vigesimal numeral system are those of Maya and Dzongkha (the national language of Bhutan). The numeral systems of the Ainu language of Japan and Kaire language of Sudan also rely, to an extent, on base 20 for the representation of numbers. Apart from Yorùbá, other African languages with vigesimal numeral system are: Madingo, Mundo, Logone, Nupe, Mende, Bongo, Efik, Vei, Igbo, and Affadeh (Conant 1896). The study by Conant (1896) highlighted the extent of the mental computation required in the expression and conception of the Yorùbá numerals, and concluded that the Yorùbá numeral system is the most peculiar numeral system in existence. One might then begin to wonder why the Yorùbá language, with a simple syllabic structure, will use such a complex numeral system. The reason for this may not be too clear.

Johnson (1921) conducted an analysis of the Yorùbá numerals by focusing on the derivation processes and the morphophonological rules required, and showed how large numbers are calculated in multiples of 20,000. The study by Abraham (1958) examined the arithmetic skills employed in different Yorùbá numeral groups, and provided a guide into their syntactic representation. A profound study on Yorùbá numerals was done by Ẹ́kúndayọ̀ (1977), where the derivational breakdown of the Yorùbá numerals was discussed and the structural representation of Yorùbá numerals was illustrated. In the study, 16 basic number lexemes which serve as the basic building blocks of the Yorùbá numeral system were identified as presented in Section 2.1.

2.1 *Basic numbers in Yorùbá*

The Yorùbá counting system has lexemes for basic numbers from 1 to 10 and six higher numerals (i.e.: 20, 30, 200, 300, 400, and 20,000). These 16 basic number lexemes are:

> *ọ̀kan* (1), *èjì* (2), *ẹ̀ta* (3), *ẹ̀rin* (4), *àrún-ún* (5), *ẹ̀fà* (6), *èje* (7), *ẹ̀jọ* (8), *ẹ̀sán-án* (9), *ẹ̀wá* (10), *ogún* (20), *ọgbọ̀n* (30), *igba* (200), *ọ̀dúnrún* (300), *irínwó* (400), *ọ̀kẹ́* (20,000) (Ẹ́kúndayọ̀ 1977)

Abraham (1958) and Ẹ́kúndayọ̀ (1977) also highlighted another set of basic numerals which are multiples of 20 from 20 to 80. These include:

> *okòó* (20), *òjì* (40), *ọ̀tà* (60), and *ọ̀rìn* (80).

These forms of lexemes are used with multiples of 100 between 200 and 20,000. The lexical representation of 20 has two values, i.e., *ogún* or *okòó*, which are used in different contexts. *Okòó* is the only form used in initial word positions when it is added to (*ó lé*) or subtracted from (*ó dín*) a vigesimal, while *ogún* is used with the multiplication formatives in numerical derivation. To illustrate this, 220 may either be expressed as *igba ó lé ogún* (200 increased by 20) or *okòólérúgba* (20 added to 200) but not as *igba ó lé okòó* or *ogúnlérúgba* although they would represent the same quantity.

Numbers are generated using syntactic combination of these lexemes, and only three of the basic mathematical operators are required to represent an infinite set of numbers within the Yorùbá language. These operators are represented by special position words like *lé* for addition, *dín* for subtraction, and *ọ̀nà* for multiplication. However, it should be pointed out that subtraction has an unusually higher functional load than addition. An exponential represented as *ìlọpo* may be required to express very large numerals as powers of 20,000 (Ọdẹ́jọbí 2003) but this is not generally used in the Yorùbá numeral system.

2.2 *Overcounting in Yorùbá numerals*

We have mentioned the use of three of the standard arithmetic operations (i.e., multiplication, subtraction, and addition) in the Yorùbá numeral system. However, it is important to discuss a special mode of subtraction depicted by *ẹẹdín* and its variant, *aadín*. The *ẹẹdín* phenomenon was well articulated in Ẹ́kúndayọ̀ (1977), where a detailed

explanation of this concept was given. Overcounting (Menninger 1969) occurs when a numeral is expressed in relation to a higher numeral. Overcounting, thus, becomes inevitable within any numeral system employing subtraction operation in number representation.

In the Yorùbá numerals system, when *ẹẹdín* is used with a number, it implies that the number must be reduced by a certain value. The use of *ẹẹdín* or *aadín* is context-dependent; hence, the value deducted varies depending on the numeral to which it is attached. This is shown in Table 2. When *ẹẹdín*' is used with numbers 20 and 30, 5 is deducted from them to produce 15 (*ẹẹdín ogún* = *ẹ̀ẹ̀dógún*) and 25 (*ẹẹdín ọgbọ̀n* = *ẹ̀ẹ̀dọ́gbọ̀n*) respectively. But if *ẹẹdín* is used with 600, 100 is deducted to produce 500 (*ẹẹdín ẹgbẹ̀ta* = *ẹ̀ẹ̀dẹ́gbẹ̀ta*).

Variant	**Number**	**Reduction**
ẹẹdín	20 and 30	5 (half of 10)
aadín	60, 80, ..., 200	10 (half of 20)
ẹẹdín	600, 800, ..., 2000	100 (half of 200)
ẹẹdín	4000, 6000, ..., 20000	1000 (half of 2000)

Table 2: '*ẹẹdin*' mode of subtraction

The concept of overcounting is also noticeable in the numeral systems of Ainu and Maya. Danish, an essentially Germanic language, also exhibits a related subtractive phenomenon (Conant 1896) as illustrated below:

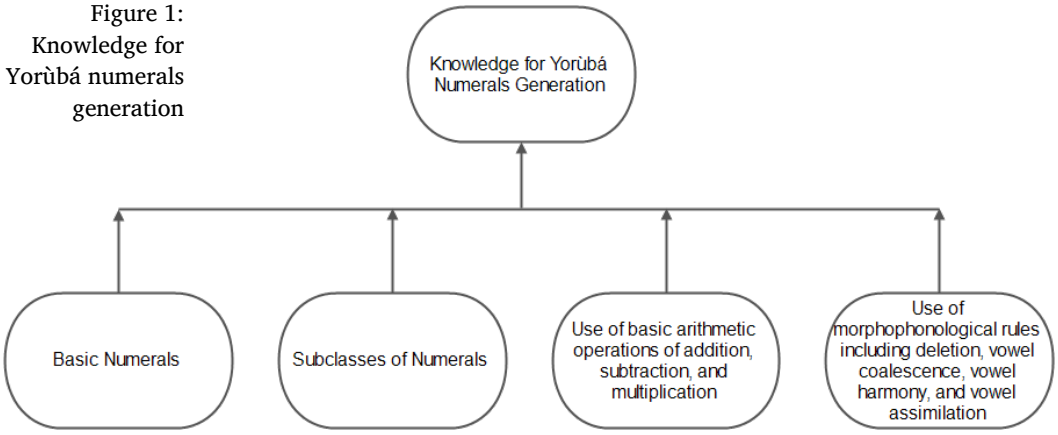
a) 50 = *halvtredsindstyve* = half (of 20) from 3 × 20
b) 70 = *halvfierdsindstyve* = half (of 20) from 4 × 20
c) 90 = *halvfemsindstyve* = half (of 20) from 5 × 20

Notably, the process of naming numbers in Danish is similar to Yorùbá. Now, we present the rules used in naming numbers in Yorùbá.

2.3 *Yorùbá number naming rules*

There are basic rules that hold in the generation of an infinite set of number names in the Yorùbá language as captured in Figure 1. As observed by Hurford (2001), numeral sequences in human languages show several discontinuities in their patterns of representation. Therefore, it is important to identify numeral groups that exhibit similar derivative process within the Yorùbá numeral system. This is to

Figure 1: Knowledge for Yorùbá numerals generation



achieve the design of an effective computational model to handle the mathematical and syntactic structure of each group. The groups are:

a) **Basic numbers:** The canonical lexemes in the Yorùbá language have been discussed in Subsection 2.1. This set of lexemes cannot be broken down to simpler forms, and other number names are generated using arithmetic combinations of these lexemes.

b) **Numbers from 11 to 200:** The addition operation is used for deriving numbers from one to four above multiples of 10, while numbers from five to one below such points are obtained through subtraction as illustrated in Figure 2. The Yorùbá lexical representation of number 11 is formed as an additive concatenation of the terms for numbers 1 and 10. This also applies to numbers 12, 13, and 14 as :

   i) $11 = (1+10) =$ *ọ̀kan lé ẹ̀wá* $=$ *ọ̀kànlá*
   ii) $12 = (2+10) =$ *èjì lé ẹ̀wá* $=$ *èjìlá*
   iii) $13 = (3+10) =$ *ẹ̀ta lé ẹ̀wá* $=$ *ẹ̀tàlá*
   iv) $14 = (4+10) =$ *ẹ̀rìn lé ẹ̀wá* $=$ *ẹ̀rìnlá*

   Note that the lexical representation of '+ 10', i.e., *lé ẹ̀wá* is contracted to *lá*. The Numbers from 15 to 19 are represented as 5 to 1 deducted from 20, respectively.

   i) 15 = 5 from 20 = *àrún-úndínlógún*
   ii) 16 = 4 from 20 = *ẹ̀rindínlógún*
   iii) 17 = 3 from 20 = *ẹ̀tadínlógún*

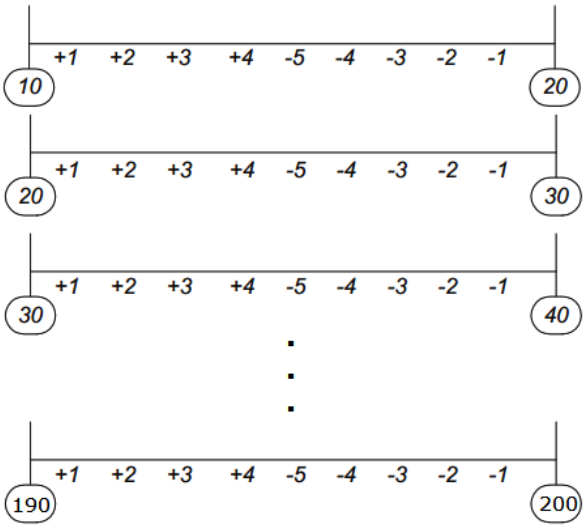


Figure 2: Yorùbá number scale

iv) 18 = 2 from 20 = *èjìdínlógún*

v) 19 = 1 from 20 = *ọ̀kàndínlógún*

Multiples of 20 from 40 to 180 are expressed as such in successive elision and vowel harmony. Numerals 50, 70, 90, 110, 130, 150 and 170 are expressed as 10 deducted (*aadín*) from the next multiple of 20. Again, this is illustrated below:

i) 40 = (20 × 2) = *ogún èjì* = *ogójì*

ii) 50 = 10 less (20 × 3) = *aadín (ogún ẹ̀ta)* = *ààdọ́ta*

iii) 110 = 10 less (20 × 6) = *aadín (ogún èfà)* = *ààdọ́fà*

iv) 180 = 20 × 9 = *ogún ẹ̀sàn-án* = *ọgọ́sàn-án*

Notably, a possible representation of 30 is *ààdóji*, which means 10 deducted from 2 twenties ((20 × 2) − 10), but 30 is referred to as *ọgbọ̀n*, which is a generic term in the Yorùbá numeral system.

c) **Numbers From 200 to 2000:** Apart from 400, numbers which are multiples of 200 are derived in multiples of *igba* and the numbers 500, 700, 900, 1100, 1300, 1500, 1700, and 1,900 are derived by 100 deducted (*ẹ̀ẹ̀dín*) from the next multiple of 200.

i) 600 = (200 × 3) = *igba ẹ̀ta* = *ẹgbẹ̀ta*

ii) 1000 = (200 × 5) = *igba àrùn-ún* = *ẹgbẹ̀rún-ún*

iii) $1400 = (200 \times 7) =$ *igba èje* = *egbèje*

iv) $1500 = (1600 - 100) = (200 \times 8) - 100 =$ *ẹẹdín igba èjọ* = *èèdégbèjọ*

v) $2000 = (200 \times 10) =$ *igba èwá* = *ẹgbèwá* = *ẹgbàá*

d) **Numbers Between 2,000 and 20,000:** Numbers in this subgroup are formed from 2,000 as the root word. The multiples of 2,000 within this range are expressed as multiples of *egbàá* and intermediate numbers are formed with the *eedín* that shows a subtraction of 1,000.

i) $6{,}000 = (2{,}000 \times 3) =$ *ẹgbàá èta* = *ẹgbààta*

ii) $10{,}000 = (2{,}000 \times 5) =$ *ẹgbàá àrún-ún* = *ẹgbààrún-ún*

iii) $15{,}000 = (16{,}000 - 1000) = (2{,}000 \times 8) - 1000 =$ *eedín ẹgbàá èjọ* = *ẹẹdẹgbààjọ*

iv) $20{,}000 = (2{,}000 \times 10) =$ *ẹgbàá èwá* = *ẹgbààwá*. This number is also expressed as *ọ̀kẹ́ kan*.

e) **Numbers 20,000 and above:** Numerals greater than 20,000 are derived as a multiple of 20,000 (*ọ̀kẹ́ kan* = twenty thousand in one place).

i) $40{,}000 = (20{,}000 \times 2) =$ *ọ̀kẹ́ méjì*

ii) $1{,}000{,}000 = (20{,}000 \times 50) =$ *àádọ́ta ọ̀kẹ́*

iii) $800{,}000{,}000 = (20{,}000 \times 20{,}000 \times 2) =$ *ọ̀kẹ́ ona ọ̀kẹ́ méjì*

iv) $8{,}000{,}000{,}000{,}000 = 20{,}000 \times 20{,}000 \times 20{,}000 =$ *ọ̀kẹ́ ọ̀nà ọ̀kẹ́ ọ̀nà ọ̀kẹ́ kan*

Once the number groups are identified, Yorùbá numerals can be represented as a combination of members from each group using the addition and subtraction operations. For example, the number 45,678 will be represented as:

$$45{,}678 = 40{,}000 + 5{,}000 + 600 + 70 + 8 \tag{1}$$

A close observation of these groups shows that certain numbers occur as reference points in the Yorùbá numeral system as proposed by Pollmann and Jansen (1996). An observable trend is that the numbers 20 and 10 play important roles within the Yorùbá numeral system.

2.4 The linguistic structure of numerals

In this section, we review two important bibliographic references on the syntactic structure of numerals. The first one is Hurford (1975), which is an extensive study of various numeral systems. The other one is the study conducted by Ẹkúndayọ̀ (1977), in which phrase structure rules were proposed for the Yorùbá numeral system.

2.4.1 Hurford's generative numeral grammar

A notable work on the application of generative grammar to numerals is the work of Hurford (1975), where the universal phrase structure rules for deriving numerals were presented. A modified version of the phrase structure rules was presented in Hurford (2007), being a significant improvement with respect to well-formed numerals. In this extensive study of numerals, Hurford considered numerals as syntactic structural constructs and proposed a universal constraint on numerals, which he called the packing strategy. The packing strategy helps to make the right choice for a number name from different structural constructs derived by the production rules presented in Definition 1. The packing strategy guides the general constraints on the well-formed nature of numerals and any structure containing an ill-formed structure is itself ill-formed.

Definition 1 (Hurford's production rules for Yorùbá numerals) *Hurford's production rules for the Yorùbá numeral system are as follows:*

$$NUM \rightarrow \left\{ \begin{array}{c} DIGIT \\ NP \end{array} \right\} (NUM) \tag{2}$$

$$NP \rightarrow (M)\ NUM \tag{3}$$

$$M \rightarrow 10\left(\left\{ \begin{array}{c} 2 \\ M \end{array} \right\} \right) \tag{4}$$

Where DIGIT is a set of basic number lexemes, M is a set of multiplicative base lexemes, NUM is a numeral and the start symbol, and NP is a Number Phrase. Rule (2) is interpreted as addition/subtraction, and it can occur in reverse order, i.e., NUM → NUM NP. Rules (3) and (4) are interpreted as multiplication when two constituents are chosen. The curly brace in the production rules shows alternative productions, while parenthesis indicates

an optional item. For example, an NP can be formed from a single NUM or a multiplicative combination of M NUM.

Hurford's generative framework provides an adequate account for the numeral system of most languages including English. However, the grammar proposed for the Yorùbá numeral system was structurally inadequate. It is worth noting that the grammar does not provide an adequate mechanism to differentiate between the addition and subtraction operations in Rule (2). For example, Hurford (1975) presented structures for 46 and 4,600, as shown in Figure 3. In the structure in Figure 3(a), 46 (*ẹ̀rindínlààdọ́ta*) was derived by deducting 4 (*ẹ̀rin*) from 50 (*ààdọ́ta*) and 50 was derived by deducting 10 from 60 (*ògọ́ta*). In Figure 3(b) and (c), representing structures for 4,600, i.e., *ẹgbẹ̀talélogún* (200 × 23) and *ẹgbàájì ó lé ẹgbẹ̀ta* (4,000 + 600) respectively, 23 (*ẹ̀talélogún*) was derived by adding 3 (*ẹ̀ta*) to 20 (*ogún*) and 4,600 was derived by 4,000 plus 600. Rule *NUM → NUM NP* is interpreted as subtraction in Figure 3(a), whereas, it is interpreted as addition in Figures 3(b) and (c). This means that the structure in Figure 3(a) could be misinterpreted as 54, and structures in Figure 3(b) and (c) as 3,400. Therefore, this introduces ambiguity in interpretation. It is also important to point out that Rule (4) results in an incorrect interpretation of the structures of *M*. To illustrate this, the rule represents 20 (*ogún*) as a combination of 10 (*ẹ̀wá*) and 2 (*ẹ̀jì*), which is structurally incorrect. This is because *ogún* is not formed, by any means, from the combination of *ẹ̀wá* and *ẹ̀jì*.

The study also acknowledged that multiple structures may exist for some numbers like 4,600, as shown in Figures 3(b) and 3(c), but concluded that the structure in 3(c) was ill-formed, whereas it is a valid structure in Yorùbá. This conclusion could result from a limited expert knowledge in verifying the correctness of these structures, as noted:

> *"Despite the difficulty in finding crucial information in the sources, it is conceivable that some complete account of Yorùbá numerals can be given that is soundly motivated. This language certainly presents the weightiest challenge for a general theory of numerals that we have encountered." (Hurford 1975, p. 232)*

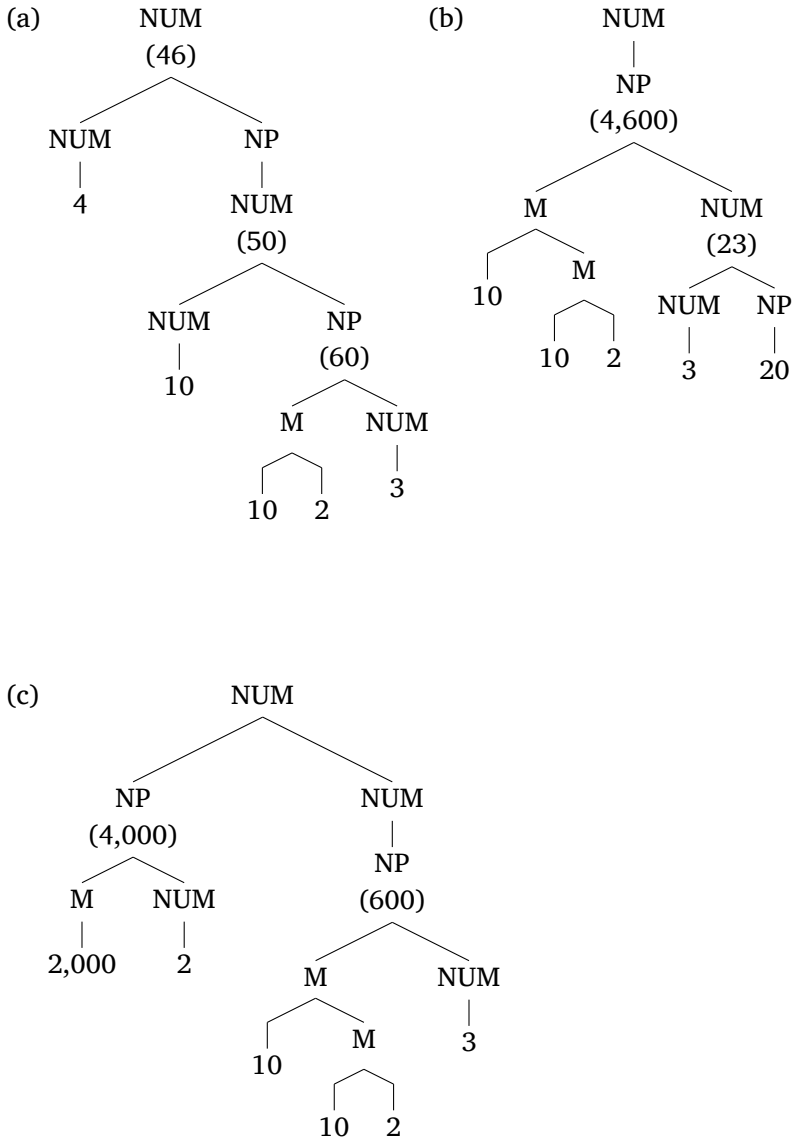


Figure 3:
Parse tree for
(a) 46 derived by 4 deducted from (10 deducted from (20×3))
(b) 4,600 derived by $200 \times (3 + 20)$
(c) 4,600 derived by $(2{,}000 \times 2) + (200 \times 3)$

2.4.2 Ẹkúndayọ̀'s phrase structure rules

The study conducted by Ẹkúndayọ̀ (1977) reveals that there exist similarities between the mechanism used in the Yorùbá language for constructing an infinite number of sentences from a finite set of building blocks and constructing an infinite set of numerals from a limited set of basic numbers. This proposition was corroborated into 3 different concepts as shown in Table 3. This shows that all Yorùbá numerals can be sententially represented through the addition, subtraction, and multiplication operators. The study also shows that some numbers have multiple representations in the Yorùbá language, but constraints of correctness are imposed on these representations. These constraints include linguistic and structural plausibilities.

Apart from the concept of infinity, creativity, and paraphrasable representation of numerals, Ẹkúndayọ̀ (1977) demonstrated that a recursive grammar is needed for numeral derivation and representation. It was observed that the recursive rules are not easily established for the Yorùbá numerals, however, a set of phrase structure (PS) rules for the Yorùbá numeral system was given as shown in Definition 2.

Table 3: Comparison of sentence construction and number naming in Yorùbá

No.	Concept	Language	Yorùbá numerals
1	**Infinity**	There is no longest sentence. Any sentence, however long, can be expanded. So with the use of recursive rules, an infinite number of sentences can be constructed.	Numerals are infinitely enumerable. This means that there is no longest numeral. Any numeral, however large can still be increased. So, the concept of recursive rules can be adopted in numerals.
2	**Creativity**	It is possible to construct and perceive an entirely new sentence that has never been heard before.	Yorùbá numerals also require a high level of creativity as higher numerals must be recreated every time they are used.
3	**Paraphrase**	A single idea could be represented in several ways.	A single number may also be represented in different forms in Yorùbá numerals.

Definition 2 (Ẹkúndayọ̀'s PS rules for Yorùbá numerals)
Ẹkúndayọ̀ phrase structure rules for the Yorùbá numeral system are as follows:

$$NUM \rightarrow NP \tag{5}$$

$$NP \rightarrow \left\{ \begin{array}{l} NP\ \ S \\ N \\ PRON \end{array} \right\} \tag{6}$$

$$S \rightarrow NP\ \ VP \tag{7}$$

$$VP \rightarrow V\ \ NP \tag{8}$$

Where NUM is a numeral and the start symbol, NP is a noun phrase, VP is a verb phrase, S is a sentence, N is the set of 16 basic number lexemes, $PRON$ is the formative ó ('it'), and V is a verb represented as the operating formatives ọ̀nà (for multiplication), dín (for subtraction), and lé (for addition). NOTE: Rule (8) was presented as $V \rightarrow V\ \ NP$ in the original article but it was modified to make the grammar complete.

The point of interest here is that verbs are used in number naming, and that numbers are sententially represented in their surface structure. This allows for a distinction between addition and subtraction operations. This is illustrated by the structure of *ẹ̀rinlélàádọ́ta* (54), shown in Figure 4, where the operating formative (V) is explicitly represented. Although these PS rules proved useful in the derivation of Yorùbá numerals, they are mostly arithmetic rather than syntactic rules as the positions of the basic lexical numerals and operatives do not correspond to their positions in the surface structure. An example would be the surface structure of *ẹ̀rinlélàádọ́ta* (54) represented in Figure 4 as ((*ọ̀gún ọ̀nà mẹ́ta) ó dín ẹ̀wá) ó lé ẹ̀rin* rather than *ẹ̀rin ó lé (aadín (ọ̀gún ọ̀nà mẹ́ta*)), thereby leading to a misrepresentation of numerals.

Another problem with Ẹkúndayọ̀'s PS rules is that multiplicative bases (M) in Hurford's grammar are not captured. The multiplicative bases help to understand which numbers are important milestones in a numeral system. Hence, in this paper, we used knowledge from these two models to capture the essential components of the Yorùbá numerals. The grammar developed captures the multiplicative bases and treats Yorùbá numerals as both arithmetic and syntactic constructs.

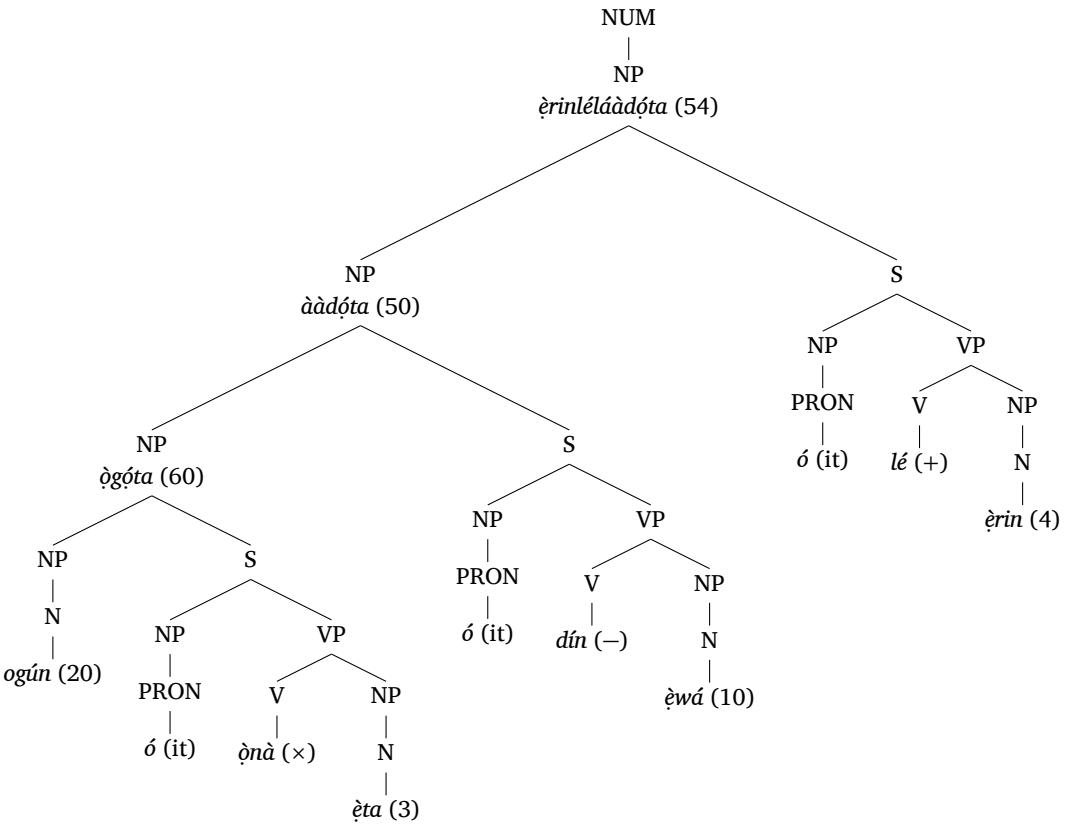


Figure 4: Parse tree for *ẹ̀rinlélàádọ́ta* (54)

3 SYSTEM DESIGN AND IMPLEMENTATION

It has been shown that the Yorùbá numeral system is very methodical, thus, an efficient computational system is required to gain accuracy in number representation. Figure 5 presents the block diagram of number transcription in the Yorùbá language. There are four important processes in this model. First, there is the number decomposition process, where numbers are expressed as a sum of smaller numbers in harmony with the sub-grouping discussed in Section 2.3. The output of this process is the magnitude stack. Next, there is a process that generates the possible forms of a single number. This is done by careful combinations of neighbouring elements of the magnitude stack and parsing them with the designed numeral grammar. This is done by using

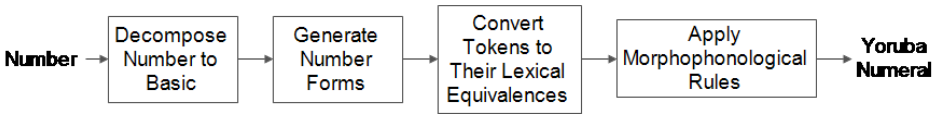


Figure 5: Number to Yorùbá text transcription system. The figure shows the processes involved in converting a cardinal number to Yorùbá text.

the packing strategy to verify whether the structures are well-formed. The third process is where tokens of the number forms are converted to their equivalent lexical forms, and the final process is where the morphophonological rules employed in Yorùbá naming numbers are applied.

3.1 *Number decomposition to vigesimal*

Within the Yorùbá numeral system, every number can be represented using a combination of five different smaller terms, each drawn from the possible groups of the Yorùbá numeral system. So, the first process is to generate the magnitude stack from the given number. This generates five new numbers $(d_0, d_1, d_2, d_3, d_4)$ from the given number. So that

$$number = d_4 + d_3 + d_2 + d_1 + d_0 \tag{9}$$

where

a) d_0 is 0 or a member of subgroup (a), i.e.,
d_0 takes values from 0 to 9, i.e., $d_0 \in DIGIT = \{0, 1, 2, ..., 9\}$.

b) d_1 is 0 or a member of subgroup (b), i.e.,
d_1 is a multiple of 20 ($d_1 = 20 \times n \mid 0 \leq n < 10$) **or** 10 deducted from a multiple of 20 ($d_1 = (20 \times n) - 10 \mid 2 \leq n \leq 10$).

c) d_2 is 0 or a member of subgroup (c), i.e.,
d_2 is a multiple of 200 ($d_2 = 200 \times n \mid 0 \leq n < 10$) **or** 100 deducted from a multiple of 200 ($d_2 = (200 \times n) - 100 \mid 2 \leq n \leq 10$).

d) d_3 is 0 or a member of subgroup (d), i.e.,
d_3 is a multiple of 2,000 ($d_3 = 2{,}000 \times n \mid 0 \leq n < 10$), **or** 1,000 deducted from a multiple of 2,000 ($d_3 = (2{,}000 \times n) - 1{,}000 \mid 2 \leq n \leq 10$).

e) d_4 is 0 or a member of subgroup (e), i.e.,
d_4 is a multiple of 20,000 ($d_4 = 20000 \times n \mid 0 \leq n < \infty$).

Table 4: Magnitude stack of some numbers

Number	Magnitude stack
23	[20, 3]
167	[160, 7]
3,459	[3,000, 400, 50, 9]
19,669	[19,000, 600, 60, 9]
412,987	[400,000, 12,000, 900, 80, 7]
1,876,234	[1,860,000, 16,000, 200, 30, 4]

These new numbers can be derived using Algorithm 1. Any of d_4, d_3, d_2, d_1, d_0 that is equal to zero is removed from the magnitude stack. The magnitude stacks of some numbers are presented in Table 4. For example, the magnitude stack generated for number 1,876,234 was:

$$[d_4, d_3, d_2, d_1, d_0] = [1{,}860{,}000,\ 16{,}000,\ 200,\ 30,\ 4]$$

In the next section, we discuss how the representations of single numbers are generated.

3.2 *Generating forms of a number*

Once the magnitude stack has been computed, the next task is to generate the possible forms of the number in Yorùbá. All the possible Yorùbá forms of a number are derived by some combinations of neighbouring elements of the magnitude stack. The possible forms for a number with magnitude stack of $[d_4, d_3, d_2, d_1, d_0]$ are listed in Table 5. For example, the magnitude stack for 19,669 is $[d_3, d_2, d_1, d_0] = [19{,}000,\ 600,\ 60,\ 9]$, and the possible forms are shown in Table 6. All possible forms are then stored in the form stack. However, not all numbers exhibit all these forms. The number of forms largely depends on the values of d_4, d_3, d_2, d_1, and d_0.

Thereafter, the elements of the form stack are expanded to a form containing only the symbols representing the basic lexical items. The expanded form stack for number 19,669 is presented in Table 7. In these forms, '×' represents multiplication, '—' and '+' represent subtraction and addition within a number phrase respectively; '——' and '++' represent subtraction and addition between number phrases respectively, as discussed in Section 3.3 b(ii). It should be noted that

Algorithm 1: Magnitude generator algorithm

```
Data: number: Input number
Result: magnitudeStack: The magnitude stack
procedure GenerateMagnitude(number)
    d_0, d_1, d_2, d_3, d_4 = 0;
    divisor = 10;
    magnitudeStack = [ ];
    while number ≠ 0 do
        remainder = number % divisor;
        if remainder ≠ 0 then
            magnitudeStack.push(remainder);
        end if
        number = number − remainder;
        divisor = divisor × 10;
    end
    for mag in magnitudeStack do
        if mag < 10 then
            d_0 = d_0 + mag;
        else if mag < 200 then
            d_1 = d_1 + mag;
        else if mag < 2000 then
            d_2 = d_2 + mag;
        else if mag < 20000 then
            d_3 = d_3 + mag;
        else
            d_4 = d_4 + mag − (mag % 20000);
            d_3 = d_3 + (mag % 20000);
        end if
    end
    magnitudeStack = [d_0, d_1, d_2, d_3, d_4];
    return magnitudeStack.reverse();
end procedure
```

arithmetic is mostly done from right to left in the Yorùbá numeral system, i.e., $2-20$ implies 2 removed from 20, which gives 18. In the same way, $(10-(20\times4))$ implies 10 deducted from (20×4) to give 70.

Table 5: Forms of Yorùbá number

	Derivation
1	$[d_4,d_3,d_2+d_1,d_0]$
2	$[d_4,d_3,d_2+d_1+20,d_0-20]$
3	$[d_4,d_3,d_2+d_1-20,d_0+20]$
4	$[d_4,d_3,d_2,d_1+d_0]$
5	$[d_4,d_3,d_2+100,d_1+d_0-100]$
6	$[d_4,d_3+1000,d_2-1000,d_1+d_0]$
7	$[d_4,d_3+1000,d_2-1000+100,d_1+d_0-100]$
8	$[d_4,d_3+1000,d_2+d_1-1000,d_0]$
9	$[d_4,d_3+d_2,d_1+d_0]$
10	$[d_4,d_3+d_2+100,d_1+d_0-100]$

Table 6: Generation of the forms of 19,669. Item d_4 is discarded because $d_4 = 0$

	Derivation	Form Stack
1	$[d_3, d_2+d_1, d_0]$	[19,000, 660, 9]
2	$[d_3, d_2+d_1+20, d_0-20]$	[19,000, 680, −11]
3	$[d_3, d_2+d1-20, d_0+20]$	[19,000, 640, 29]
4	$[d_3, d_2, d_1+d_0]$	[19,000, 600, 69]
5	$[d_3, d_2+100, d_1+d_0-100]$	[19,000, 700, −31]
6	$[d_3+1000, d_2-1000, d_1+d_0]$	[20,000, −400, 69]
7	$[d_3+1000, d_2-1000+100, d_1+d_0-100]$	[20,000, −300, −31]
8	$[d_3+1000, d_2+d1-1000, d_0]$	[20,000, −340, 9]
9	$[d_3+d_2, d_1+d_0]$	[19,600, 69]
10	$[d_3+d_2+100, d_1+d_0-100]$	[19,700, −31]

3.3 *Context-free grammar for Yorùbá numerals*

We studied the structures of the five numeral groups discussed in Section 2.3, from which some patterns became apparent. We started the design of the CFG by identifying the set of terminal symbols which are:

a) The set of lexemes listed in Section 2.1.

   i) *DIGIT* = {*ọ̀kan* (1), *èjì* (2), *ẹ̀ta* (3), *ẹ̀rin* (4), *àrún-ún* (5), *ẹ̀fà* (6), *èje* (7), *ẹ̀jọ* (8), *ẹ̀sán-án* (9), *ẹ̀wá* (10), *ọgbọ̀n* (30), *ọ̀dún-rún* (300), *irínwó* (400), *okòó* (D20), *òjì* (D40), *ọ̀tà* (D60), *ọ̀rìn* (D80)}, and

   ii) The set of multiplicative bases i.e., M = {*ogún* (20), *igba* (200), *ọ̀kẹ́* (20,000)}

b) The sets of lexical affixes depicting arithmetic operations in Yorùbá numerals. These three sets of operators are:

No	Form Stack
1	$[(1{,}000-(2{,}000\times 10))++(D60+(200\times 3))++9]$
2	$[(1{,}000-(2{,}000\times 10))++(D80+(200\times 3))--(1+10)]$
3	$[(1{,}000-(2{,}000\times 10))++(D40+(200\times 3))++(1-30)]$
4	$[(1{,}000-(2{,}000\times 10))++(200\times 3)++(1-(10-(20\times 4)))]$
5	$[(1{,}000-(2{,}000\times 10))++(100-(200\times 4))--(1+30)]$
6	$[20{,}000--400++(1-(10-(20\times 4)))]$
7	$[20{,}000--300--(1+30)]$
8	$[20{,}000--(D40+300)++9]$
9	$[(200\times(2-(20\times 5)))++(1-(10-(20\times 4)))]$
10	$[(100-(200\times(1-(20\times 5))))--(1+30)]$

Table 7: Expanded forms of number 19,669. '++' and '––' are operation formatives found between phrases, while *D*40, *D*60, and *D*80 are multiples of 20 as mentioned in Section 2.1

i) A set of operators that occur within a phrase. This includes *lé ní* (+) for addition and *dín ní* (−) for subtraction. Multiplication within a phrase is implied, which means it is not explicitly represented. Hence, $V = \{lé\ ní,\ dín\ ní\}$.

ii) A set of operators that occur between phrases. This includes *ó lé* (++) for addition, *ó dín* (––) for subtraction and *ọ̀nà* (×) for multiplication. So we say $VV = \{ó\ lé,\ ó\ dín,\ ọ̀nà\}$.

iii) A set of implied subtraction operators represented by the prefixes *aadín* (reduction by 10) and *ẹẹdín* (reduction by 5, 100, and 1,000), i.e., $REDUCE = \{aadín,\ ẹẹdín\}$.

Thus the set of terminal symbols, T, is made up of all elements in: *DIGIT*, M, V, VV, and *REDUCE*.

The start symbol is a numeral which is denoted by *NUM*. Since a CFG is the union of simpler grammars (Sipser 2007), we started by constructing rules for structures of numerals that could occur as a number phrase.

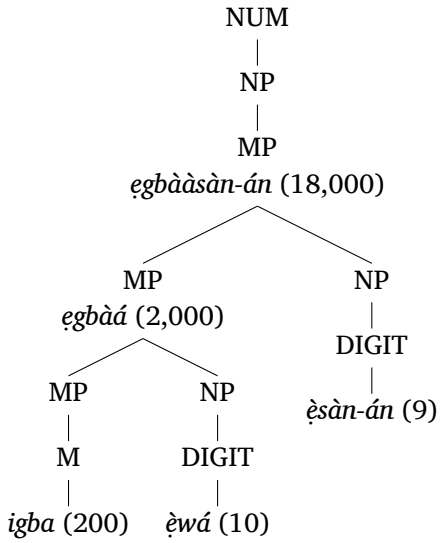
A phrase could be formed as a single *DIGIT* (Ẹkúndayọ̀ 1977) or from the multiplication of M and *DIGIT*. A phrase formed by multiplication is denoted by *MP* (Hurford 2007), i.e.,

$$NP \rightarrow DIGIT \mid MP \tag{10}$$

$$MP \rightarrow M \mid MP\ NP \tag{11}$$

MP is formed by a single multiplicative base ***M***, or recursively by multiplying *MP* by a number phrase *NP*, e.g. *ọgọ́ta* is formed by multiplying an *MP* (20 – formed by $MP \rightarrow M$) and an *NP* (3 – formed

Figure 6:
Parse tree of
ẹgbàásàn-án (18,000)



by *NP → DIGIT*). Also, Rule (11) is recursive to handle multiple levels of multiplication. For example, 18,000 (*ẹgbàásán*) is represented as 2,000 multiplied by 9, and 2,000 is subsequently represented as 200 multiplied by 10, as shown in the parse tree in Figure 6.

We then added a rule to make allowance for the *ẹẹdín/aadín* type of subtraction. The phrase *ẹẹdín/aadín* can only occur as a prefix to a number derived from a multiplication operation. When this is done, the value deducted depends on the number to which it is prefixed (discussed in Subsection 2.2). A further example is 50 (*àádọ́ta*), which is derived by deducting 10 from 60 (*ọgọ́ta*). Rule (12) captures this as shown in the structure in Figure 7.

$$NP \rightarrow REDUCE\ MP \tag{12}$$

With the inclusion of this rule, it should be pointed out that it has some obvious consequences. The rule overgenerates, that is, it allows the use of *ẹẹdín* or *aadín* without respecting Table 2. We shall devise means of filtering out ill-formed structures using the packing strategy.

The next stage refers to how the operators *V* (Verbs) are represented within a phrase. Within a phrase, the Yorùbás start number presentation with the smaller number (Addend/Subtrahend) rather than the larger number (Augend/Minuend). For instance, number 21

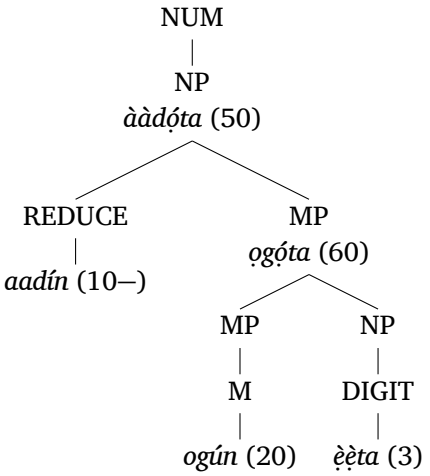


Figure 7:
Parse tree of *ààdọ́ta* (50)

(twenty one) is represented as *ọ̀kànelélógún* (1+20) in Yorùbá. We then considered '1 +' as a verb phrase (*VP*), which is made up of a *DIGIT* and a *V* as presented in Rule (13):

$$VP \rightarrow DIGIT \;\; V \tag{13}$$

A *VP* can then be combined with an *NP* to make up an *NP*, i.e.:

$$NP \rightarrow VP \;\; NP \tag{14}$$

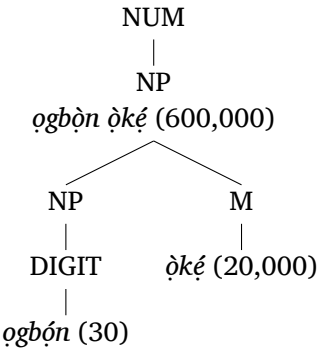
Also, the order in Rule (11) can be reversed to capture the structure of numbers like 600,000 (*ọgbọ̀n ọ̀kẹ́*), which is represented as 30 times 20,000. The multiplicative base is now positioned at the end of the rule, and can only take *ọ̀kẹ́* (20,000) as a value. The outcome of this new rule is a phrase (*NP*), since it cannot be used as a multiplicand (*MP*) to derive higher numerals. For example, 1,200,000 cannot be represented as *ọgbọ̀n ọ̀kẹ́ ọ̀nà mẹ́wàá* ((30 × 20,000) × 10), but as *ọ̀dúnrún ọ̀kẹ́* (300 × 20,000). So we added Rule (15). The structure of number 600,000 is shown in Figure 8.

$$NP \rightarrow NP \; M \tag{15}$$

Next, we created rules to connect these phrases together to form a number. So, a number could be formed from a phrase, i.e.:

$$NUM \rightarrow NP \tag{16}$$

Figure 8:
Parse tree of
ọgbọ̀n ọ̀kẹ́ (600,000)



Also, a number could be formed by combining an existing number with a phrase using the lexical operatives in the set *VV*. We added two rules to capture this as follows:

$$NUM \rightarrow NUM\ S \quad (17)$$

$$S \rightarrow VV\ NP \quad (18)$$

Although multiplication plays an important role in Yorùbá numerals, its lexical representation, *ọ̀nà* does not occur in number names except when more than one 20,000 (*ọ̀kẹ́*) occur within a number phrase. For example, 400,000,000 is represented as 20,000 × 20,000, i.e., *ọ̀kẹ́ ọ̀nà ọ̀kẹ́ kan*, and the structure is also captured using Rule (18) as shown in Figure 9.

Finally, all these rules were merged to make up the production rules of the Yorùbá numeral grammar, as presented in Definition 3.

Definition 3 (Production rules of the Yorùbá numeral grammar) *The production rules for the Yorùbá numeral system are as follows:*

$$NUM \rightarrow NP \mid NUM\ S \quad (19)$$

$$S \rightarrow VV\ NP \quad (20)$$

$$NP \rightarrow DIGIT \mid MP \mid VP\ NP \quad (21)$$

$$NP \rightarrow REDUCE\ MP \mid NP\ M \quad (22)$$

$$MP \rightarrow M \mid MP\ NP \quad (23)$$

$$VP \rightarrow DIGIT\ V \quad (24)$$

These phrase structure rules include the verbs which are operating formatives (V and VV) proposed by (Ẹ́kúndayọ̀ 1977). These rules pro-

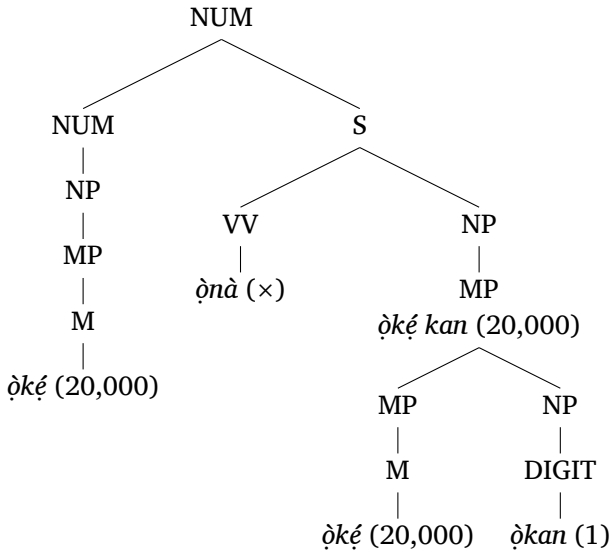


Figure 9:
Parse tree of
ọ̀kẹ́ ọ̀nà ọ̀kẹ́ kan
(400,000,000)

duce a single and correct structure for most Yorùbá numerals, however, the rules overgenerate with some numerals. For example, the number 1,000,000 produces 3 structures as presented in Figure 10, but the valid structure is determined using a single packing strategy defined in Definition 4.

Definition 4 (Packing strategy for the Yorùbá numeral system)
The following metarules govern well-formed Yorùbá numeral structures:

- *(i) Whenever a phrase MP is formed by a multiplicative combination of two numerals, the multiplicand (MP) must be greater than the multiplier (NP).*
- *(ii) Whenever the rule NP → REDUCE MP is used, the lexical item of REDUCE must correspond to the appropriate MP, as shown in Table 2.*
- *(iii) Whenever the rule S → VV NP is used and the VV has the value of ọ̀nà, then NP can only take a value of ọ̀kẹ́, and the resulting S must be used with a multiple of ọ̀kẹ́. (see Figure 9).*

Using the packing strategy for Yorùbá numerals, the well-formedness of the structures in Figure 10 was investigated and only the structure in (c) was well-formed. The analysis is as follows:

(i) In the structure in Figure 10(b), the formation of *MP* from *ogún* (20) disagrees with metarule (i), thereby making the structure in Figure 10(b) ill-formed.

(ii) The structure in Figure 10(a) is not well-formed as *REDUCE* (*aadín* (10—)) is applied to a multiple of *ọ̀kẹ́* (20,000), which

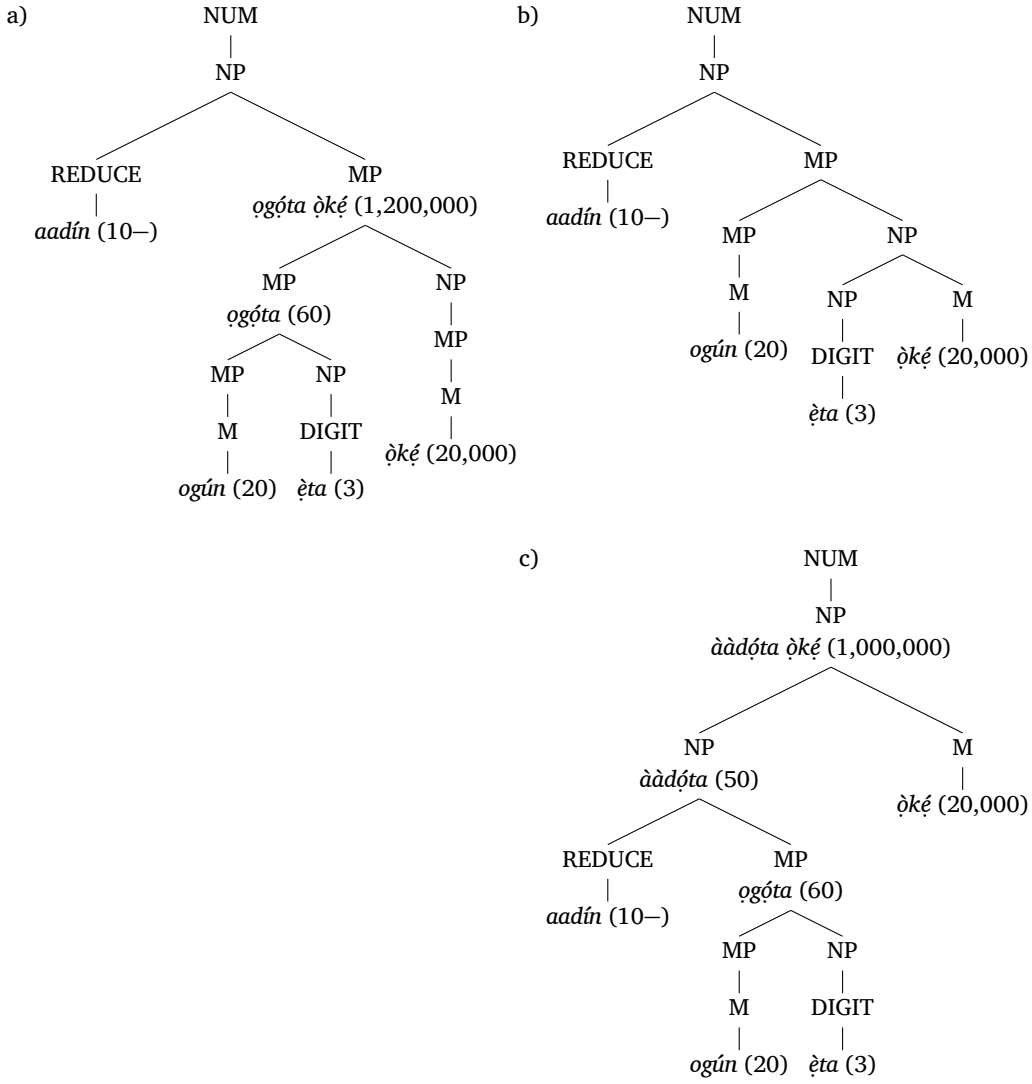


Figure 10: Parse trees of *ààdọ́ta ọ̀kẹ́* (1,000,000)

disagrees with metarule (ii). Hence, only the structure in Figure 10(c) is well-formed.

Once a parse tree is generated, we then convert the tokens to their lexical equivalences followed by the application of morphophonological rules.

3.4 *Morphophonological rules in Yorùbá numeral system*

The representation of numbers in Yorùbá is cumbersome due to the fact that a high level of linguistic processing is involved. Therefore, the speakers are required to have adequate knowledge of some morphophonological rules in the Yorùbá language. These morphophonological rules include deletion, vowel coalescence, vowel harmony, and vowel assimilation. These rules will be discussed to show how they are useful in number naming.

3.4.1 Deletion

Deletion is a process by which a phrase or word is shortened by completely deleting a segment. Both vowels and consonants can be deleted in Yorùbá. The most commonly deleted consonants are *w* (when it is part of the last syllable) and *g*. Deletion is notable in the contracted form of phrases *dín ní* (less than) and *lé ní* (more than), where *i* is completely deleted and *n* is converted to *l*. This conversion is possible because *n* and *l* are allophones of the same phoneme. For example, the expression for 28, which is derived as 2 from 30 (*èjì dín ní ọgbọ̀n*), is *èjìdínlógbọ̀n*.

A deletion also occurs in naming numbers between 11 and 14. For example, *ókànlá* (11) is formed by adding 1 to 10, i.e., *òkan lé ẹ̀wá*, which is contracted to form *òkanlẹ́wá* by deleting the vowel *é*. The consonant *w* and vowel *ẹ* are then deleted to form *òkànlá*. Another example is *ẹ̀ẹ̀dégbẹ̀ta*, which is formed from *ẹẹdín ẹgbẹ̀ta*. This is achieved by completely deleting the vowel *ín*.

3.4.2 Vowel coalescence

Coalescence is a phonological process whereby two adjoining segments converge or fuse into one element such that the new segment is

phonologically distinct from the input segments (Bámiṣilẹ̀ 1994). This is illustrated by Equation 25, where V_1 is the vowel that ends the first morpheme, V_2 is the vowel that begins the second morpheme, '+' is the morpheme boundary, and V_3 is the resulting morpheme.

$$V_1 + V_2 \rightarrow V_3 \tag{25}$$

In coalescence, the combining vowels may be phonologically distinct from each other but the resulting vowel must be distinct from the combining vowels, i.e., $V_1 \neq V_3$ and $V_2 \neq V_3$ (Awóbùlúyì 1987).

Vowel coalescence is most notable when two nouns are next to each other. And since Yorùbá numerals are mostly treated as nominal entities, they also use vowel coalescence in naming numbers. For example, vowel coalescence is used in the formation of *ogójì* (40) derived from 20 multiplied by 2, i.e., *ogún èjì*. The vowels *ún* and *e* are combined by coalescence to become *o*. Table 8 shows the possible occurrence of vowel coalescence in the Yorùbá numeral system.

Table 8: Vowel coalescence in Yorùbá numerals. V_1 is the vowel that ends the first morpheme, V_2 is the vowel that begins the second morpheme, + is the morpheme boundary, and → stands for 'rewritten as'

V_1	V_2	V_3	**Example**
ún	*à*	*ọ́*	*ogún + àrùn-ún → ọgọ́rùn-ún*
ún	*è*	*ọ́*	*ogún + èjì → ọgọ́jì*
ún	*ẹ̀*	*ọ*	*ogún + ẹ̀ta → ọgọ́ta*
í	*i*	*ú*	*èjì dín ní + igba → èjìdínlúgba*

3.4.3 Vowel harmony

Standard Yorùbá has 7 oral vowels, which are: *a, e, ẹ, i, o, ọ, u*. Vowel harmony places a constraint on the occurrence of vowel sequence in words. Archangeli and Pulleyblank (1989) discussed the two classes of Standard Yorùbá oral vowels which are:

i) Advanced Tongue Root (ATR), which are vowels *i, e, o,* and *u*,
ii) Non-ATR, which are vowels *a, ẹ*, and *ọ*

ATR vowels involve drawing forward the root of the tongue so that the pharynx is expanded. In simple Yorùbá words, the last vowel in the word determines the other vowels in the word (Akinlabí 2004). So, if the last vowel in a word is an ATR, the immediately preceding vowel must be an ATR. The high vowels (*i* and *u*) do not participate

in the vowel harmony at all, and they can occur with any vowel. Only the mid vowels (*e, o, ẹ*, and *ọ*) are fully involved in the vowel harmony (Akinlabí 2004). The chart presented in Table 9 shows the permissible and non-permissible sequences of vowels in the Yorùbá language.

		V_2						
		i	e	ẹ	a	ọ	o	u
V_1	i	+	+	+	+	+	+	+
	e	+	+	Ø	Ø	Ø	+	+
	ẹ	+	Ø	+	+	+	Ø	+
	a	+	+	+	+	+	+	+
	ọ	+	Ø	+	+	+	Ø	+
	o	+	+	Ø	Ø	Ø	+	+
	u*	+	+	+	+	+	+	+

Table 9:
Sequence of vowels in Yorùbá bisyllabic words. The symbols + and Ø indicate the permissible and non-permissible vowel sequence, respectively. V_2 is the second oral vowel in the word and V_1 indicates the vowel that may precede V_2

(Adapted from Archangeli and Pulleyblank (1989))
*The letter u cannot start a word in the Standard Yorùbá language.

These rules also apply to number naming in Yorùbá as illustrated with the following example: The number *ọgọ́ta* (120) is derived as 20 (*ogún*) multiplied by 3 (*ẹ̀ta*), i.e., *ogún ẹ̀ta*. The vowels *ún* and *ẹ̀* are then changed to vowel *ọ́* to form *ogọ́ta* by means of vowel coalescence. Since the last vowel in the last two syllables is *a*, which is a non-ATR, therefore, the immediately preceding vowels must be non-ATR. We will then proceed to check for harmony between the first two syllables. The second vowel *ọ́* is non-ATR, therefore, the first vowel *o* must also be a non-ATR. This will transform *o* to *ọ* by means of the vowel harmony.

3.4.4 Vowel assimilation

Vowel assimilation is a process whereby a vowel becomes completely or partially like another vowel (Akinlabí 2004). Vowel assimilation is most notable in Yorùbá numerals when a consonant separating 2 vowels is deleted. This can be illustrated by number 2,000 (*ẹgbàá*). The number 2,000 is actually formed from 200×10, i.e., *igba ẹ̀wá*, which will produce *ẹgbẹ̀wá* by vowel deletion. *ẹgbàá* is then formed by deleting the consonant *w* and allowing the vowel *ẹ* to assimilate the form of vowel *a*.

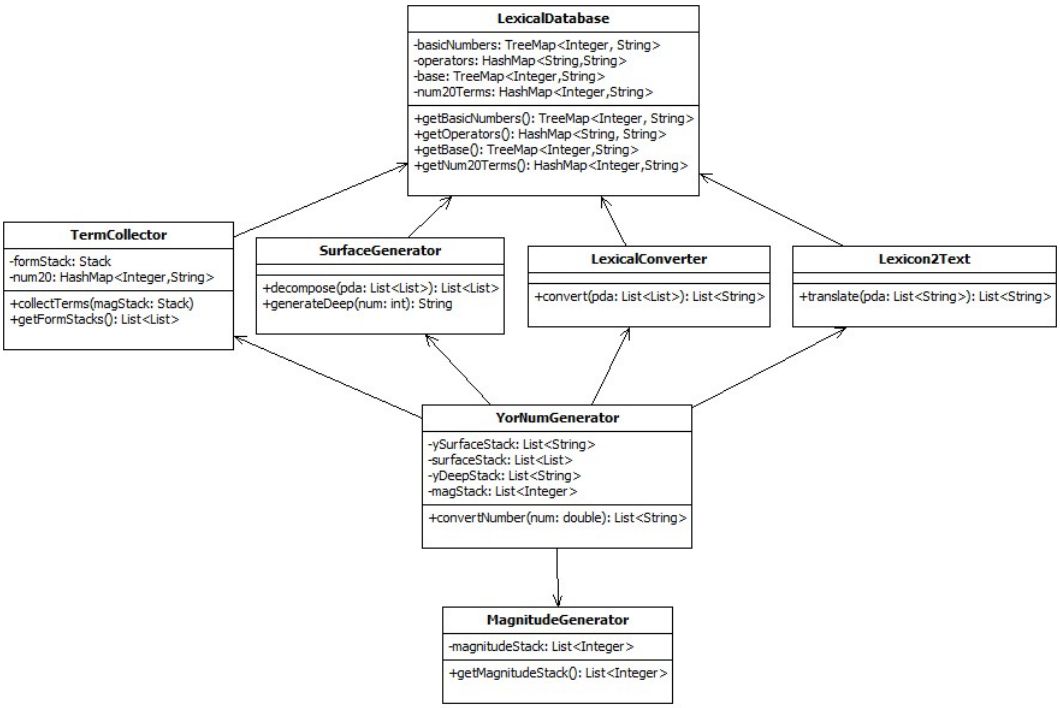


Figure 11: UML class diagram

Vowel assimilation can also occur between vowels separated by a consonant as in the expression for 800 (*ẹgbẹ̀rin*). This expression is derived as 200 (*igba*) multiplied by 4 (*èrin*), i.e., *igba èrin*. *igbẹ̀rin* is then formed by deleting *a*. The vowel *i* then assimilates *ẹ* to form *ẹgbẹ̀rin*.

3.5 *System and implementation*

An object oriented programming (OOP) approach with 7 classes was used during the system design. The UML class diagram and the sequence diagram for the software are as shown in Figure 11 and Figure 12 respectively.

The software implementation was done using Python and Java. The software was implemented following the specifications in the system design. The following software pieces were developed to demonstrate the conversion of numbers to Standard Yorùbá text:

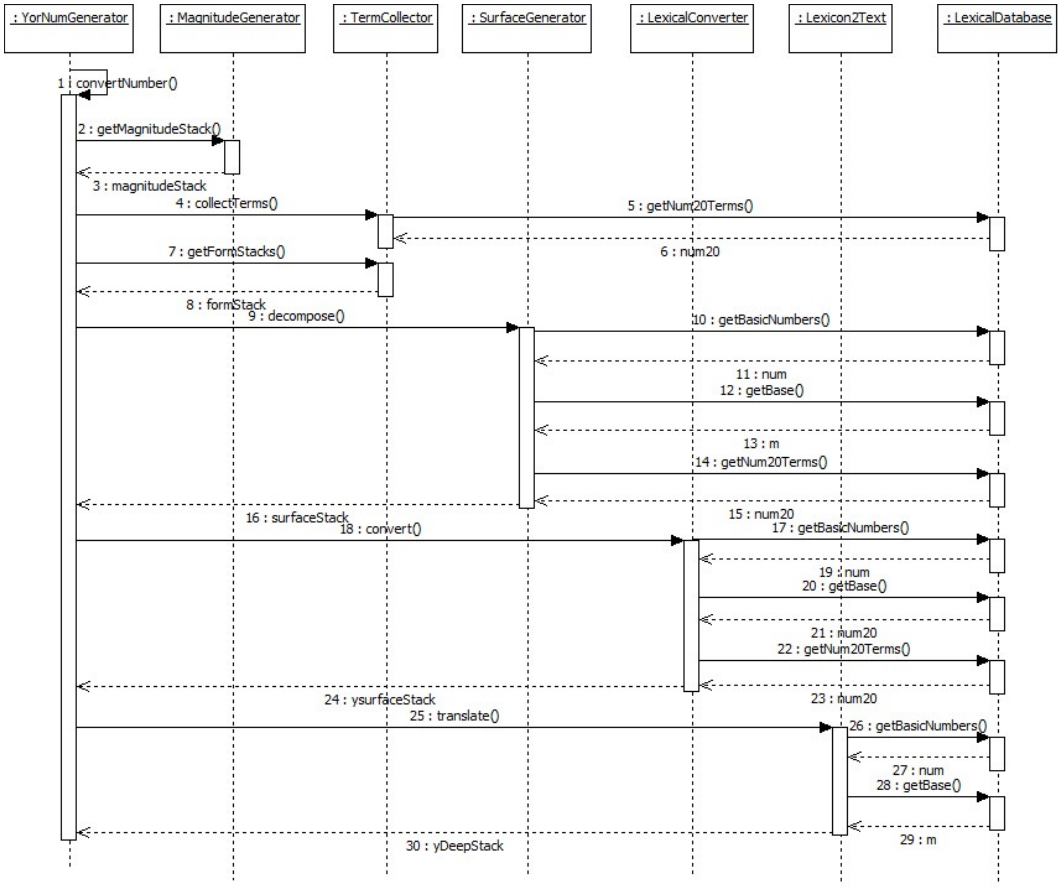


Figure 12: UML sequence diagram

a) **Desktop application:** The desktop application was implemented using PyQt in the Python programming language environment. The combination of Python and Qt makes possible the development of applications that are platform-independent (Summerfield 2008). NLTK (Loper and Bird 2002) was used to implement the grammar designed for the Yorùbá numeral system. It was also used to generate the parse trees of the number forms. The screenshot is as shown in Figure 13 and the software is available for download at `http://www.ifecisrg.org/yorubanumerals`.

b) **Web application:** The web application was implemented using the Google App Engine Python API. The screenshot is as shown in

Figure 14. The application is available at `http://www.num2yor.appspot.com`

c) **Mobile application on Android OS:** The mobile application was ported to Android using Java and the Android Application Development Toolkit (ADT). The screenshot is as shown in Figure 15.

The desktop application has a single document interface with toolbars for all tasks on top, a menu bar duplicating toolbar tasks, and a dockable history and analysis widgets. The analysis widget shows the computational details of a numeral structure. The Onka software has the following features:

a) The history can be saved for future usage.

b) Users can copy the output text to the computer's clipboard and paste it into an editing program or word processor.

c) The output of the software can be printed or saved in Unicode text format.

d) LaTeX users can copy or save the output in the LaTeX format. Also, the parse trees generated can be copied in the qtree (Siskind and Dimitriadis 2008) bracketed syntax for inclusion in TeX documents.

4 DISCUSSION

The software produces the correct lexical transcription for numbers in the Yorùbá language. In the following subsections, analysis will be carried out on the structure, computation, and forms of certain numbers. The numbers that will be considered are 240, 969, 19,669, and 40,000,000.

4.1 *The number 240*

The software processing of 240 produced two different forms, which are:

a. *òjìlélígba*: This number is computed by the addition of digit 40 to 200, i.e., [D40 + 200]. This representation uses only one addition operation. The parse tree of this representation is shown in Figure 16. This representation contains three terminal symbols, and the depth of the parse tree is 6.

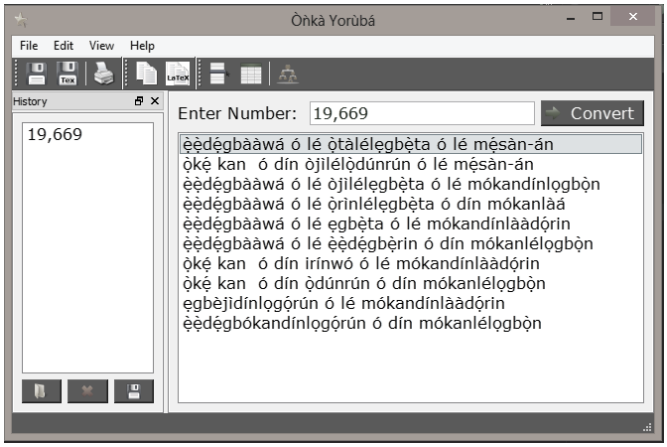


Figure 13:
Screenshot of Òǹkà desktop application

Figure 14:
Screenshot of Òǹkà web application hosted on Google App engine

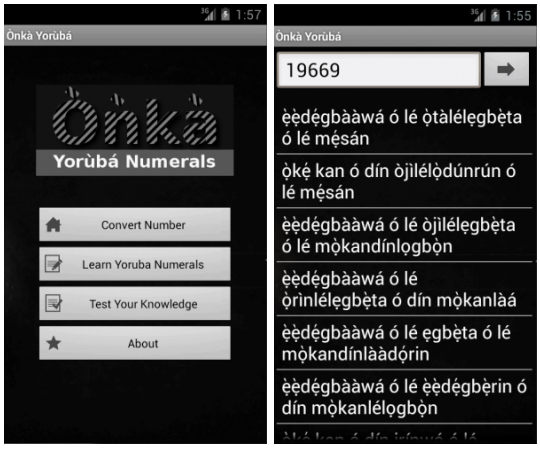


Figure 15:
Screenshot of *Òǹkà* Android application

Figure 16:
Parse tree of *òjìlélígba* (240 = 40 added to 200) – representation 1

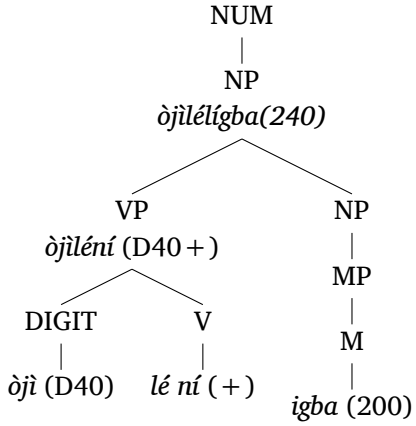
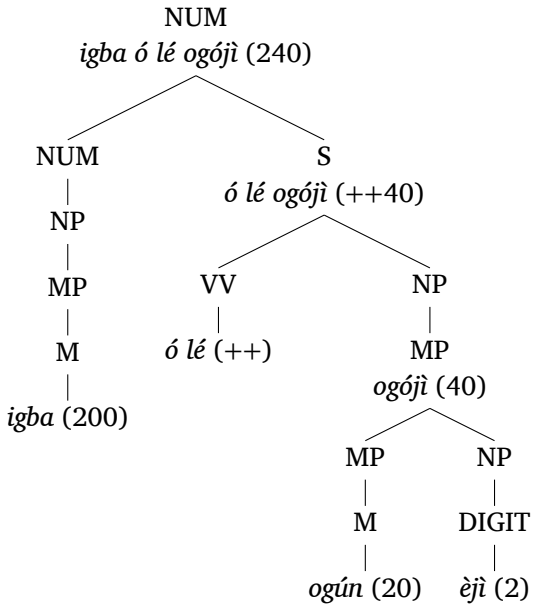


Figure 17:
Parse tree of *igba ó lé ogójì* (240 = 200 increased by 40) – representation 2



b. *igba ó lé ogójì*: This representation is presented as a phrase containing two number phrases. The parse tree of this representation is shown in Figure 17. This representation contains four terminal symbols, and the depth of the parse tree is 7.

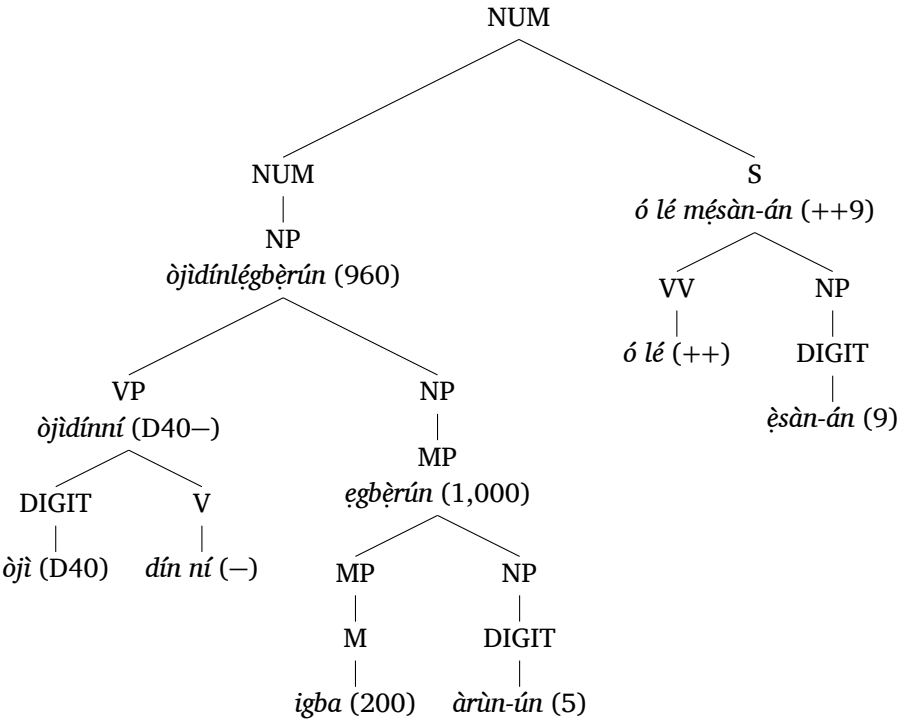


Figure 18: Parse tree for *òjìdínlẹgbẹ̀rún ó lé mẹ́sán* (969)

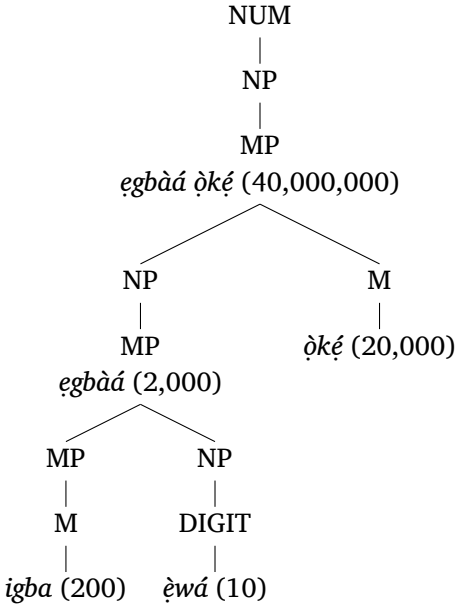
4.2 *The number 969*

The software gives 5 representations for number 969 as shown in Table 10. All these representations are valid and none has preference over others. The choice of a representation depends on the mental dexterity of the speaker. The parse tree for the first representation is shown in Figure 18.

Table 10: Representations of number 969

No	Representation	Derivation
1	*òjìdínlẹ́gbẹ̀rún ó lé mẹ́sàn-án*	$((200 \times 5) - 40) + 9$
2	*ọ̀tàdínlẹ́gbẹ̀rún ó lé mókandínlogbọ̀n*	$((200 \times 5) - 60) + (-1 + 30)$
3	*okòódínlẹgbẹ̀rún ó dín mókanlàá*	$((200 \times 5) - 20) - 11$
4	*ẹ̀ẹ̀dégbẹ̀rún ó lé mókandínlààdọ́rin*	$((200 \times 5) - 100) + (((20 \times 4) - 10) - 1)$
5	*ẹgbẹ̀rún ó dín mókanlélogbọ̀n*	$(200 \times 5) - (1 + 30)$

Figure 19:
Parse tree of *ẹgbàá ọ̀kẹ́* (40,000,000)



4.3 *The number 19,669*

The output of the software for number 19,669 is shown in Figure 14. Representations 1 to 7 were presented by Ẹkúndayọ̀ (1977) and the developed software produced three more representations (8–10) that are structurally valid.

4.4 *The number 40,000,000*

The software gave one representation for 40,000,000 (*ẹgbàá ọ̀kẹ́*), which is derived as a multiple of 20,000 (i.e., $2{,}000 \times 20{,}000$). Next, 2,000 was derived as 200 in 10 places, i.e., 200×10. The parse tree is shown in Figure 19.

5 SYSTEM EVALUATION

In order to determine the accuracy of the system, we analysed and evaluated the output generated using the qualitative evaluation method. However, in these circumstances, it becomes expedient to rank the output of the software when multiple representations are produced. The aim is to order the representations according to the economy of computation.

5.1 *Number name ranking*

Although all representations produced are valid, we proposed some heuristic measures for ranking the representations when there are multiple correct expressions for a number. Once the parse tree had been generated for each representation, we computed the values to determine the computational economy of the numeral structure in the following order:

i) **The total number of terminal nodes (t):** This represents the number of basic lexical items that make up a Yorùbá numeral. The fewer the number of terminal nodes, the more economical the numeral structure is.

ii) **The height of the parse tree (h):** The height of the generated parse tree was determined by using the *height()* function of the package *nltk.tree*. The parse tree with the least height is thus considered the most suitable representation for a number.

iii) **The relative number of subtractions (r):** The most natural operations in most numeral systems are addition and multiplication, yet, the Yorùbá numeral system places a higher functional load on subtraction.
The value of r is calculated by dividing the number of subtraction operations by the total number of arithmetic operations as shown in Equation 26. The two possible types of subtraction are the normal subtraction operation and the *ẹẹdín* type of subtraction.

$$r = \frac{\textit{Number of subtraction operations}}{\textit{Total number of arithmetic operations}} \tag{26}$$

This means that a lower r implies a higher economy.

Once the first measure has been calculated and some structures have the same cost, the second measure, which checks the height of the parse tree in each structure, is used. But, if there is still a tie in values among any of the structures, the last measure (i.e., the relative number of subtraction) is used to determine the most suitable representation for a number. To illustrate this, we used these measures to decide which of the two representations for the number 240 discussed in Section 4.1 is more computationally economical. We started

by picking the representation with the minimum number of terminals. The parse tree in Figure 16 has three terminal symbols compared to four in Figure 17. Thus, the structure in Figure 16 is more computationally economical.

Also, the analysis of the ten representations for the number 19,669 is presented in Table 11. This shows the number of terminal symbols, the depth of the parse tree, and the arithmetic complexity. The computational cost was calculated based on these criteria, and it was used to rank the representations. The representations with the lowest number of terminal symbols and least height are representations 2 and 8 (with 8 terminal symbols and height of 8), however, representation 2 has the lesser relative number of subtractions. Hence, representation 2 (Figure 20) is the most computationally economical. Table 12 presents the most economical representations of selected numbers derived from the software.

Table 11: Representations for the number 19,669 and their ranks. **t** is the number of terminal symbols, **h** is the height of the parse tree as generated by the software, and **r** is the relative number of subtraction operations in the representation

No	Rank	Yorùbá Text	t	h	r
2	1	*ọkẹ ó dín ojìlélọ́dúnrún ó lé mẹ́sán-án*	8	8	0.250
8	2	*ọkẹ ó dín ọ́dúnrún ó dín mọ́kanlélọ́gbọ̀n*	8	8	0.500
7	3	*ọkẹ ó dín irínwó ó lé mọ́kandínlaadọ́rin*	10	8	0.500
10	4	*ẹẹ́dẹ́gbọ̀kandínlọ́gọ́rún ó dín mọ́kanlélọ́gbọ̀n*	10	10	0.500
1	5	*ẹẹ́dẹ́gbàáwá ó lé ọ̀talélẹ́gbẹ̀ta ó lé mẹ́sán*	11	9	0.143
9	6	*ẹ́gbèjìdínlọ́gọ́rún ó lé mọ́kandínlaadọ́rin*	11	10	0.429
6	7	*ẹẹ́dẹ́gbàáwá ó lé ẹẹ́dẹ́gbẹ̀rin ó dín mọ́kanlélọ́gbọ̀n*	12	9	0.375
3	8	*ẹẹ́dẹ́gbàáwá ó lé ójìlélẹ́gbẹ̀ta ó lé mọ́kandínlọ́gbọ̀n*	13	9	0.250
4	9	*ẹẹ́dẹ́gbàáwá ó lé ọ̀rinélẹ́gbẹ̀ta dín mọ́kanlàá*	13	9	0.250
5	10	*ẹẹ́dẹ́gbàáwá ó lé ẹ́gbẹ̀ta ó lé mọ́kandínlaadọ́rin*	13	9	0.333

5.2 *Qualitative evaluation*

The Mean Opinion Score (MOS) was used for the qualitative evaluation of the system. Chosen members of the staff of Ọbáfẹ́mi Awólọ́wọ̀ University, who are Yorùbá native speakers with adequate knowledge of the Yorùbá language and its orthography, were asked to provide the textual equivalences of some numbers in Yorùbá. Afterwards, their responses were compared to the output from the software.

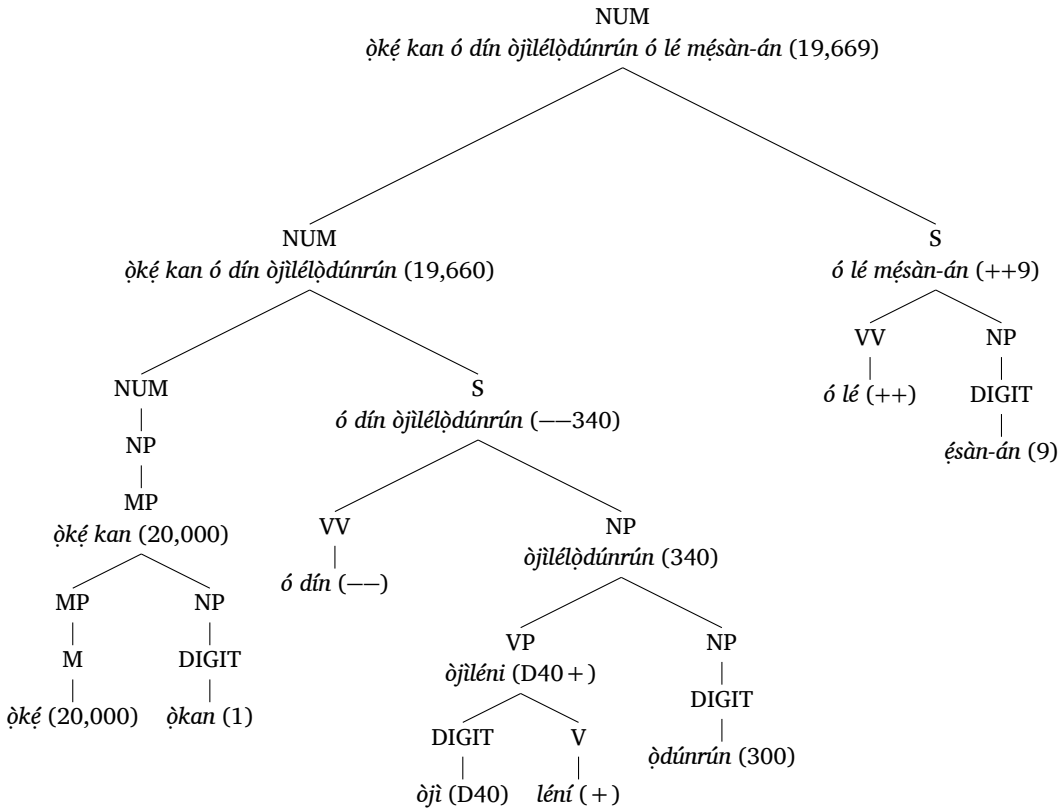


Figure 20: Parse tree for *ọ̀kẹ́ kan ó dín ọ̀jìlélọ̀dúnrún ó lé mẹ́sàn-án* (19,669)

Table 12: Software output for some numbers

Number	**Yorùbá Text**
182	*ọgọ́sán ó lé méjì*
187	*ọgọ́sán ó lé méje*
365	*irinwó ó dín márùndínlógójí*
595	*ẹgbẹ̀ta ó dín márùn-ún*
666	*ọ̀tàlélẹgbẹ̀ta ó lé mẹfà*
760	*ọ̀jìdínlẹ́gbẹ̀rin*
777	*ẹ̀ẹ̀dẹ́gbẹ̀rin ó lé mẹ́tàdínlógọ́rin*
815	*ẹgbẹ̀rin ó lé márùndínlógún*
840	*ọ̀jìlélẹ́gbẹ̀rin*
905	*ẹ̀ẹ̀dẹ́gbẹ̀rún ó lé márùn-ún*
1,247	*ọ̀jìlélẹ́gbẹ̀fà ó lé méje*
600,000	*ọgbọ̀n ọ̀kẹ́*

A questionnaire was designed and administered to the selected group of 32 respondents. The numbers in the questionnaire were 25, 67, 132, 750, 969, 2,400, 3,000, 19,669, 20,000, 30,000, 1,000,000, and 400,000,000. The MOS evaluation was carried out to capture two important aspects of the Yorùbá numeral system. The first one was the ability of the respondents to give an accurate representation of the numbers in terms of value and orthography, and the second one was to obtain the most suitable representation for the numbers as provided by the respondents.

The numbers used in the questionnaires were chosen based on the following criteria:

i) Numbers 25, 67, and 132 were included to confirm that numbers between 1 and 200 have one standard lexical form.
ii) Numbers 750, 969, 2,400, and 19,669 were included to check whether the respondents are aware that there are multiple representations for these Yorùbá numerals.
iii) The number 20,000 was included to check whether the respondents find 20,000 as a single lexical item or think it is derived from the number 200.
iv) Numbers higher than 20,000 (30,000, 1,000,000, 400,000,000) were included to see if the respondents represent these numbers as multiples of 20,000 or in some other way.
v) Some structurally complex numbers (969 and 19,669) were added to see the most convenient combination of basic lexical numerals used by the respondents to derive these numbers.

The results of the analysis revealed that:

- For numbers 25, 67, and 132, all the respondents gave one correct representation, which matched up with the output from the software. This shows that numbers below 200 have one standard lexical form and that the skills needed to name these numbers are well understood.
- Ten respondents gave a representation for 19,669 but only two of them gave a correct number name (*ẹẹdẹgbàáwá ó lé ọtalélẹgbẹta ó lé mẹ́sán* and *ẹẹdẹgbàáwá ó lé ẹgbẹta ó lé mọkandínlaadọ́rin*). The other eight respondents provided number names that do not in any way evaluate to the number 19,669. Twenty two (22) of the

respondents did not give any representation for number 19,669. This shows that few respondents understand that 19,669 needs to be reconstructed and only two respondents were able to carry out the required computations. This result also shows that none of the respondents realised that multiple representations exist for the number 19,669.

- Seven of the respondents gave the correct number names for 20,000, with only two respondents using *ọ̀kẹ́*, and the remaining five using *ẹgbàawá*. This shows that few respondents were able to represent 20,000 in Yorùbá.
- Only three respondents gave the correct names for 1,000,000 (*ààdọ́ta ọ̀kẹ́*), and none of the respondents gave the correct representation for 400,000,000 (*ọ̀kẹ́ ọ̀nà ọ̀kẹ́*). This shows thatYorùbá native speakers may find the computations underlying naming large numbers cumbersome.

From these results, we conclude that the respondents were able to produce correct representations for numbers that are frequently used (number 1 to 200), although most of them were not able to produce names for higher numbers. After comparing the responses of the human evaluators with the system output, we recognise that the software out-performed the human evaluators. This affirms that most native-speakers know the terminologies needed for large numbers but are not familiar with the expression skills required for computing their number names. Without a doubt, modern Yorùbá speakers are losing the numeral generation skills embedded in their language. An obvious reason for this is the overwhelming use of the English numerals within the Yorùbá community.

6 CONCLUSION

In this paper, we discussed extensively the computational analysis of the Yorùbá numerals. We started by identifying the basic lexical numerals and the numeral groups. Then, we designed a CFG that was able to capture the structure of the Yorùbá numerals. Furthermore, we implemented a software for converting numbers to their textual equivalences in the Yorùbá language and generating their corresponding parse trees.

In this study, we are able to show that:

1. The Yorùbá number system has a systematic concept underlying it and that this concept can be articulated using modern computing tools and techniques.
2. The Yorùbá numeral system is not fully vigesimal. Elements of decimal (base 10) and quinary (base 5) are used in numeral representation.
3. The system's recall is 100% with respect to the corpus used in this study. This implies that, with carefully constructed computational model, the generation of the Yorùbá numeral system can be fully automated.
4. All the forms of number names produced were valid and the most computationally suitable representation are those in which : (a) the least number of terminal nodes is used, (b) the least height of the parse tree is generated, and (c) the least relative number of subtraction operations is involved. Though these measures are computationally reasonable, an interesting study will be to verify why Yorùbá native speakers sometimes prefer to adopt more complex methods, particularly when generating numerals greater that 200.

The results of this study can be applied in Yorùbá TTS. In any TTS system, numbers must be expanded into their textual forms before the actual speech synthesis is carried out. Thus, the system developed can serve as a sub-system of a Yorùbá TTS to handle the expansion of numbers to their textual equivalences. However, additional heuristic strategies must be employed by the TTS listeners to understand the number being spoken. Without a doubt, an increased usage of the Yorùbá numerals in communication could reduce the mental task needed for number conception.

The software developed in this study has a place in effective teaching and learning of the Yorùbá language. The software can be used in classes to teach the Yorùbá numeral system and its structure. This will allow the students to see the various forms possible for a single number and to visualise the structure (parse tree) of the numerals.

There are certain areas related to this study which we cannot explore. By pointing out these areas, we hope to focus our future study on

them. There is a need to carry out the contextual analysis of the Yorùbá numeral systems which will establish the relationships between numerals and their surrounding words. This will ensure that the expansion of numbers is carried out based on the context (cardinal, ordinal, nominal, currency, percentage, ratio, date, time, etc.) they represent. Also, there is a need to carry out a study on how the textual forms of the Yorùbá numerals could be recognised and converted to numbers. Definitely, the results of these studies could be applied in Yorùbá MT and information retrieval.

ACKNOWLEDGEMENT

This work is supported by TETFUND Grant TETF/DESS/NRF/OAU/STI /VOL.1/B1.13.9. We would like to thank the editor and anonymous reviewers for their useful comments and suggestions to enhance the quality of the paper. We also acknowledge the support of the African Languages Technology Initiative (Alt-i).

REFERENCES

Wándé Abímbọ́lá (1977), *Ifa Divinity Poetry*, Traditional African Literature, Nok Pub Intl, New York.

Roy Clive Abraham (1958), *Dictionary of Modern Yorùbá*, University of London Press, London.

Akinbiyi Akinlabí (2004), *Understanding Yorùbá Life and Culture*, chapter The Sound System of Yorùbá, pp. 453–468, Africa World Press, Trenton, NJ 08607.

Diana Archangeli and Douglas Pulleyblank (1989), Yorùbá Vowel Harmony, *Linguistic Inquiry*, 20(2):173–218.

Ọládélé Awóbùlúyì (1987), Towards a Typology of Coalescence, *Journal of West African Languages*, 17(2):5–22.

David Bailey and Jonathan Borwein (2011), The Greatest Mathematical Discovery, *manuscript: Available online:* `http://escholarship.org/uc/item/0sp6t6h5`.

Rẹ̀mí Bámiṣilẹ̀ (1994), Justification for the Survival of Vowel Coalescence as a Phonological Process in Yorùbá, *African Languages and Cultures*, 7(2):133–142.

Levi Leonard Conant (1896), *The Number Concept: Its Origin and Development*, MacMillan, New York.

Samuel ẸKÚNDAYỌ̀ (1977), Vigesimal Numeral Derivational Morphology: Yorùbá Grammatical Competence Epitomized, *Anthropological Linguistics*, 19(9):436–453, http://www.jstor.org/stable/30027551.

Didier GOYVAERTS (1980), Counting in Logo, *Anthropological Linguistics*, 22(8):pp. 317–328, ISSN 00035483, http://www.jstor.org/stable/30027492.

James HURFORD (1975), *The Linguistic Theory of Numerals*, Cambridge University Press, Cambridge, ISBN 9780521133685.

James HURFORD (2001), Numeral Systems, in *International Encyclopedia of the Social & Behavioral Sciences*, pp. 10756–10761, Elsevier Science Ltd.

James HURFORD (2007), A Performed Practice Explains a Linguistic Universal: Counting Gives the Packing Strategy, *Lingua*, 117(5):773–783, doi:10.1016/j.lingua.2006.03.002, http://www.isrl.uiuc.edu/~amag/langev/paper/hurford06packingStrategy.html.

Samuel JOHNSON (1921), *The History of the Yorùbás: From the Earliest Times to the Beginning of the British Protectorate*, Routledge and Kegan Paul, London, reprinted 1966.

Edward LOPER and Steven BIRD (2002), NLTK: The Natural Language Toolkit, in *Proceedings of the ACL02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, volume 1, p. 8, http://arxiv.org/abs/cs/0205028.

Paul LOVEJOY and David TROTMAN (2003), *Trans-Atlantic Dimensions of Ethnicity in the African Diaspora*, Continuum: New York.

Adolphus MANN (1887), Notes on the Numeral System of the Yorùbá Nation, *The Journal of the Anthropological Institute of Great Britain and Ireland*, 16:59–64, available online: http://www.jstor.org/stable/2841738.

Karl MENNINGER (1969), *Number Words and Number Symbols: A Cultural History of Numbers*, MIT Press, Cambridge, translated by Paul Broneer for the revised German edition.

Ọdẹ́túnjí Àjàdí ỌDẸ́JỌBÍ (2003), Towards a Formal Specification of Some Computational Concepts in Yorùbá Thoughts, *ODU: Ifẹ̀ Journal of the Institute of Cultural Studies*, 8:87–110.

Kọ́lá OWÓLABÍ (2006), Yorùbá, *Encyclopedia of Language & Linguistics (Second Edition)*, pp. 735–738.

Thijs POLLMANN and Carel JANSEN (1996), The Language User as an Arithmetician, *Cognition*, 59:219–237.

Geoffrey SAXE (1981), Body Parts as Numerals: A Developmental Analysis of Numeration among the Oksapmin in Papua New Guinea, *Child Development*, 51(1):306–316, Blackwell Publishing on behalf of the Society for Research in Child Development.

Michael SIPSER (2007), *Introduction to the Theory of Computation*, Thomas Course Technology, India, 2^{nd} edition, ISBN 81-315-0162-0.

Jeffrey Mark SISKIND and Alexis DIMITRIADIS (2008), Qtree, a LaTeX Tree-drawing Package, Available online: `http://www.ling.upenn.edu/advice/latex/qtree/` (Accessed 19 September 2011).

Richard SPROAT (1996), Multilingual Text Analysis for Text-to-Speech Synthesis, in W. WAHLSTER, editor, 12^{th} *European Conference on Artificial Intelligence*, pp. 75–80, John Wiley & Sons, Ltd.

Mark SUMMERFIELD (2008), *Rapid GUI Programming with Python and Qt*, Prentice Hall, New Jersey, 1^{st} edition.

Helen VERRAN (2001), *Science and an African Logic*, University of Chicago Press, Chicago.

Claudia ZASLAVSKY (1973), *Africa Counts: Number and Pattern in African Cultures*, Lawrence Hill Books, 3^{rd} edition.