University of Ibadan
Journal of Science and Logics in ICT Research

A Computational Model for Studying the Characteristics of Languages using Coded Character Sets

Bamidele Oluwade[1], Adenike Osofisan[2], S. A. Ilori[3], P. B. Shola[4] and R. Akin-Ojo[5].

[1]Society for the Advancement of ICT & Comparative Knowledge, POB 20253, UIPO, Ibadan, Nigeria. deleoluwade@yahoo.com

[2]Department of Computer Science, University of Ibadan, Nigeria. nikeosofisan@gmail.com

[3]Deparment of Mathematics, University of Ibadan, Ibadan, Nigeria. ailori@yahoo.com

[4]Department of Computer Science, University of Ilorin, Ilorin, Nigeria. pbshola@yahoo.com

[5]Department of Physics, University of Ibadan, Ibadan, Nigeria. rakinojo@yahoo.com

Abstract

Computer languages possess the structure for studying all categories of languages due to the fact that all languages have basic character sets which are the fundamental building blocks for their syntaxes. This is the basis for this paper wherein a general theoretical model is presented for studying the characteristics of languages. The model hinges on the data structure of the basic character set of FORTRAN programming language when considered as a subset of three standard coded character sets which are subsets of Unicode. The model is based on the application of a method for representing binary uniform digital codes, called 'code presentation'. The focus of the paper is on the suitability of the method for representing codes, even though the method is a lossless compression algorithm. The model, for example, provides insight into whether the composition of words in a particular language belongs to that language or another. Further work may be done to establish mathematical relationship, using code presentation, to show when two languages belong to the same family.

Keywords- FORTRAN programming language, Basic character set, Code presentation, Coded character sets, Languages, Computational model

1. INTRODUCTION

All languages have the same generic structure. That is, every language has a basic character set (B) such that a concatenation of one or more elements of the set gives rise to a word or string (W). A combination of two or more words produce a phrase (P) while a combination of phrases give rise to a sentence or statement (S). The entire vocabulary (V) of a language is formed from statements. The hierarchical structure of a typical language is depicted in Figure 1.

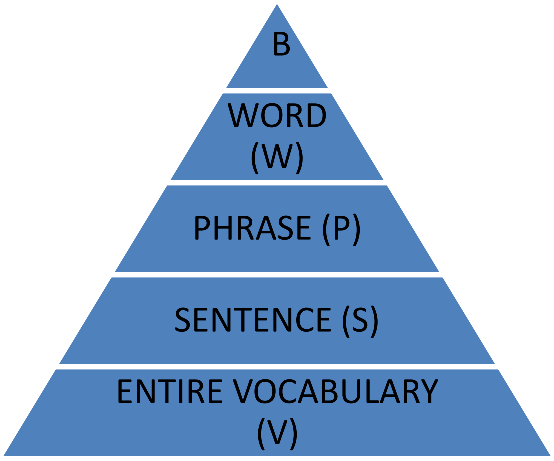

Figure 1: Basic Language Structure of Every Language (where B is the basic character set)

Bamidele Oluwade, Adenike Osofisan, S. A. Ilori, P. B. Shola and R. Akin-Ojo (2020). A Computational Model for Studying the Characteristics of Languages Using Coded Character Sets, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, Vol. 4 No. 1, pp. 85 - 95

©U IJSLICTR Vol. 4 No. 1, March 2020

Certainly, there is close affinity between computer languages and human languages. Cursorily, this may be gleaned from the historical simplification of computer languages from machine language to assembly language and then to high level language. The latter is closer to English language and facilitates easier human-computer interaction. In particular, computer languages belong to (classified) families in the same way that human languages belong to families. Several studies have been reported in the literature on general linguistic analyses such as phonetic analysis of human languages. Examples of the languages are English, Afrikaans, Zulu, Xhosa [3] and the Germanic languages [4]. Computational linguistics, in particular, attempts to understand hidden patterns and structures in languages using computational techniques. Since the basic character set is the fundamental element of a language, it is clear that it is an important concept in the language process.

Languages are potent instruments of political, economic, technological, religious and social power, and have therefore been important in world affairs from time immemorial. Historically, it is believed that there was a single (human) language in the entire world before the present-day proliferation of languages, as related in the story of the Tower of Babel in Genesis 11:1-9 [1].

Mathematically, the transition from a mono-language to several languages may be described as a one-to-many function/relation while the reverse is said to be many-to-one [8, 9]. These are well-studied concepts. Thus theoretically, it is possible to recover the single original language when a language has transmuted into several other languages.

In the present paper, a model is presented for studying the characteristics of arbitrary languages using the character sets of computer programming languages. The computer character set which is the focus of the paper is that of FORTRAN programming language. Studying a typical entire language (vocabulary) may be cumbersome since an infinite or finitely many number of words or sentences may be constructed within it. An easy way therefore is to use the (coded) character set of the language. Thus, in the present paper, a hypothetical character set of a Language I (L1) is studied within the complete vocabulary of three other hypothetical Language II (L2), Language III (L3) and Language IV (L4). Nothing is known relating to whether these languages belong to the same family.

An essential similarity between the English Language (one of the most popular human languages) and hypothetical languages L1, L2, L3 and L4 is that all are languages! Also, all the basic characters in the English Language (lower and upper case alphabetic) are in L2, L3 and L4. The basic alphabetic characters of L1 are contained in the English Language; L1 only has uppercase alphabetics. However, some differences include:

(i) The basic character set of L1 is just a subset of L2, L3 and L4.

(ii) L2, L3 and L4 are coded character sets (i.e. they have assigned meaning to every character) while the basic character set of English Language does not have assigned meanings; it's the words formed from the latter that have meanings.

(iii) The basic character set of L1 contains alphabetics, numerics and special characters while that of the English Language contains only alphabetics.

Essentially, the data structure of the basic character set of FORTRAN (herein modeled as L1) is studied such that it is considered as a subset of the 7-bit ASCII (American Standard Code for Information Interchange) [11, 12, 13, 14] (herein modeled as L2), 8-bit EBCDIC (Extended Binary Coded Decimal Interchange Code) [15, 16] (modeled as L3) and the 8-bit standard BCD (Binary Coded Decimal) [16] (modeled as L4) coding systems, where an additional odd parity check bit is appended to the least significant bit of every word in each coding system.

The paper is predicated on representation method for binary uniform digital codes, called "code presentation". This is a lossless data compression algorithm wherein a binary uniform digital code is simply described in terms of a set of 'zoned portions' and a set of 'numeric portions' in line with the traditional computer architecture [17, 18, 19, 20].

That is, the algorithm has as its foundation the conventional arrangement of words in a digital computer configuration/architecture as a zoned portion and a numeric portion [31]. It is akin to the standard method for depicting groups known as 'group presentation' in combinatorial group theory [8, 21, 22]. Although group presentation is a well developed algorithm, it doesn't take into consideration the peculiarities of a computer code, since it is strictly applicable to groups. Although all groups are codes, not all codes are groups i.e. not all codes satisfy the basic axioms of a group. Hence, there is need for a compression algorithm, specifically designed for codes, which take into consideration their peculiarities. Code presentation adequately handles these peculiarities. It turns out that it is a lossless algorithm, like the Huffman code and Lempel-Ziv method, since no data is lost in the compression process, and decompression retrieves all data that are compressed. An earlier work dwelt on the code presentation characteristics of the CCITT#2 code [23]. Details on the technique of code presentation and its areas of application have been presented in several other earlier works e.g. see [18, 19, 24, 26].

2. Literature Review

In this section, a review of some pertinent literature on the theme of the paper is presented. This includes the concepts of group codes, code presentation lossless algorithm, FORTRAN programming language and non-computer languages (i.e. human languages).

2.1 **Fortran Programming Language and Non-Computer Languages**

FORTRAN programming language was used in the present paper because it is a popular computer language out of the hundreds of programming languages that have evolved since the advent of the stored program computers (von Neumann architecture). It is one of the few old generation languages that are still relatively/fairly in use despite the emergence of several new competing languages over the years, especially object-oriented languages. For instance, it is still widely used in the fields of physics and (petroleum) engineering, including oil exploration. The language, which was originally developed in the 1950s for scientific computing, belongs to one of the four basic families of programming languages, named imperative languages [10]. This is a family of languages that is built on commands which act on stored data and modifies the overall state of the system. Other languages in this family include Visual Basic, C, postscript, PHP etc. The remaining three families are the object-oriented languages (e.g. Java, C++, C#, Simula 67), functional languages (e.g. Lisp) and declarative languages (e.g. Prolog).

There is presently about 6,000 human languages in the whole world which belong to over 100 (distinct) families of languages [2]. Table 1.1 shows some human languages and their families. These families include Indo-European, Niger-Congo and Afro-Asiatic [3, 4, 5, 6]. Nigeria as a country contributes about 500 languages out of the thousands in existence [7]. Two of the country's three major human languages, namely Yoruba and Igbo, have the same family viz. Niger-Congo family. The third major language, Hausa, belongs to the Afro-Asiatic family, a family in which Arabic is also a member.

Table 1: Some Human Languages and Their Families

S/N	HUMAN LANGUA-GE	FAMILY	EXAMPLES OF COUNTRIES WHERE THE LANGUAGE IS SPOKEN
1	English	Indo-European	United Kingdom (UK), United States of America (USA)
2	Afrikaans	Indo-European	Southern Africa e.g. South Africa
3	Yoruba	Niger-Congo	West Africa, mainly in Southwestern Nigeria and Republic of Benin
4	Hausa	Afro-Asiatic	Mainly Northern Nigeria and Southern part of Niger Republic
5	Igbo	Niger-Congo	South Eastern Nigeria
6	French	Indo-European	Mainly France and her former colonies e.g. Cameroon, Cote d'lvoire (former Ivory Coast)
7	Arabic	Afro-Asiatic	Arab countries, including Saudi Arabia, Iraq, Iran
8	German	Indo-European	Germany and some parts of Europe
9	Siwu	Niger-Congo	Ghana

In general, new languages evolve due to diffusion of people, inter-cultural marriages, migration, inter-market relationships etc. Similar reasons account for the extinction of languages. The above is partly responsible for semblance of similarity in languages which appear to be culturally distinct e.g. both the Yoruba of Southern Nigeria and Hausa of Northern Nigeria refer to onion as 'alubosa'. The inference is that the characters 'a', 'l', 'u', 'b', 'o', and 's' somehow exist in the two languages. Also, Afrikaans, a language spoken in South Africa, is technically classified as a low Franconian West Germanic language. This is due to the fact that it is a Germanic language which originates from the Dutch spoken by settlers in Africa in the 17th century C.E [4, 5].

From the sociological and political point of view, language studies indirectly help to promote world unity. For instance, the Yoruba and Igbo people of Nigeria, though have present day distinct languages, need to see themselves as brothers and sisters since they both belong to the same language family, namely Niger-Congo linguistic family. Globally, all cultures in the world need to see themselves as one since they had the same original language before the days of Tower of Babel. The understanding derivable from this knowledge is expected, at least theoretically, to minimize strive and war in the world.

Languages may be studied intellectually from the perspective of several fields in the arts and humanities. From the perspective of the sciences, the fields of mathematics and computer science are two classic examples of fields which have inherent tools for non-trivial human language studies. For instance, languages can be studied from the point of view of set theory and group theory in mathematics [47]. In computer science, its data structures can be of interest. Even though human languages are relatively inherently complex in nature, they (especially their character sets) may be studied from the perspective of computational linguistics by noting that each and every human language is somehow a subset of Unicode. [3, 25]. For instance, while the non-coded character set of FORTRAN computer language (F) contains numerics, alphabetic and special characters, the non-coded character set of English language (E) contains only alphabetic. In particular, the uppercase alphabetic of F are exactly the uppercase alphabetic of E.

The study of distinct families of languages assists in tracing the single language from where all other languages evolved. That is, language study assists in providing insight into the etymology (origin and history) of languages, settlements and communities. The starting point is the basic character sets of these languages.

2.2 **Basic Concepts of Code Presentation Compression Algorithm**

In this subsection, some of the basic definitions, concepts and results of the code presentation lossless compression algorithm are stated. These facts have been presented in several earlier works [e.g. 17, 18, 19, 20, 24, 25, 26]. Code presentation is both a representation system for binary uniform digital codes as well as a compression method. It is based on discrete structures.

This algorithm was developed for the compression of a text file consisting of a set of strings of uniform blocklength in two alphabets [18, 24]. The study of the compression characteristics of texts is important in the optimization of storage spaces. The algorithm, whose highlights are presented, has earlier been shown to have a good compression ratio [19]. In [25], the algorithm was applied to the design of new coded character sets as subsets of Unicode. It was also applied to the development of a framework for studying computer character codes based on African languages, called African Computer Character Codes. In [26], the focus was the application of code presentation to bioinformatics. The paper first presented an extension to earlier framework of code presentation. The compression algorithm was then applied to Gray code, an error detection and correction code in digital communication systems. By so doing, new characteristics of the code were presented. Thereafter, the extended nomenclature was applied to the compression of genetic sequences.

Some related works focus on the compression of texts written in specific human languages using popular text compression algorithms like the Huffman code, LZW and LZ77 [27, 28]. These human languages include Japanese, Chinese [29] and Arabic [30].

Definition 2.1

Let C be a uniform digital code of blocklength n and S a finite subset of C. Suppose $w_1 = a_1\, a_2\ldots a_p$ and $w_2 = a_{p+1}\, a_{p+2}\ldots a_n$ where $w_1, w_2 \in S$. Then the word $w = w_1w_2$ is called a juxtaposed word.

Remark 2.1

Definition 2.1 has been formalized to be in line with a related definition in combinatorial group theory [8, 22]

Definition 2.2

w_1 and w_2 are respectively defined as the zoned portion and numeric portion of w.

Definition 2.3

The block length of w_1 (i.e. p) is defined as:

$$p = \begin{cases} \frac{n}{2} & \text{if n is even} \\ \frac{n+1}{2} & \text{if n is odd} \\ \frac{n-1}{2} & \text{if n is odd} \end{cases} \qquad (2.1)$$

Definition 2.4

Based on Definition 2.3, if the block length of C is odd, two possible cases arise. The case in which $p = {}^{(n+1)}/2$ is called the Type I definition of zoned portion whilst the other case (i.e. $p = {}^{(n-1)}/2$) is called the Type II definition.

Definition 2.5

A constant zoned portion is defined as a zoned portion which is common to two or more words in C.

Definition 2.6

A subset E of C is said to be an equizone if all the words in E have the same zoned portion.

Definition 2.7

A decinumer of E is defined as the decimal equivalent of a numeric portion of the equizone whilst the order-preserving set of all the decinumers of E is called a decinumer set.

Definition 2.8

The number of equizones in C is called the degree of the code.

Definition 2.9

The numer code of C is the order preserving set of all the numeric portions of C while the zoned code of C is the order preserving set of all the zoned portions of C.

Definition 2.10

Suppose C is a uniform digital code which has a degree d. Let E_i be the equizone of decinumer set Q_i where $1 \underline{<} i < d$. Then the code presentation of C is given by

$$C = \bigcup_{i=1}^{d} \{ z_i x_{ig} \ \forall g \in Q_{i\}} \qquad (2.2)$$

where z_i is the constant zoned portion of E_i and x_{ig} the bit pattern of $g \in Q_i$.

Apart from the application of data compression algorithm to language theory, as espoused in this paper via code presentation, different compression algorithms have been applied to diverse other areas. These include power systems [36], electroencephalographic, biomedical and tele-monitoring system [37, 38], information filtering and document recommender system [39], concealment of a document within another document (steganography) [40] and big data [41]. In [42], the significance of discrete/data structures in formulating mathematical models was showcased via the development of a model for construction of components of building structures.

2.3 Some Relevant Results on the Inter-relationship between a Group and a Code

In this subsection, some basic definitions on groups and group codes are presented.

Definition 2.11 [8, 32]

Let G be a set and * the binary operation defined on it. Then (G,*) is said to be a group if it satisfies all the following four axioms, known as axioms of a group:

(i) Closure property: Let a, b ϵ G. Then a*b ϵ G for all a, b ϵ G.

(ii) Identity property: There exists an element e ϵ G such that a*e = e for all a, b ϵ G. Then e is called the identity element of G.

(iii) Inverse property: There exists an element a^{-1} ϵ G, called the inverse of a, such that a* a^{-1} = e.

(iv) Associative property: For all a, b, c ϵ G, then (a*b)*c = a*(b*c)

Definition 2.12 [33, 34]

A code is said to be a group code if it is a group.

Definition 2.13 [33, 34, 35]

A code is said to be a linear code if any linear combination of codewords that are members of the code is also a code.

Theorem 2.1 (Lagrange's Group Theorem)

A subset H of a group G is said to be a subgroup of G (written as $H \leq G$) if and only if the order of H divides the order of G.

Proof

The proof is standard and is available in virtually all basic abstract algebra and group theory texts e.g. [8, 32]. The proof uses Definition 2.11 and proceeds in two stages. First, there is need to show that if $H \leq G$, then the order of H divides the order of G. Conversely, one shows that if the order of H divides the order of G, then $H \leq G$. □

Definition 2.14 [35]

A code S is said to be linear if

$HW^t = 0$

for all code word W, where H is called the parity check matrix and W^t is the transpose of W. If otherwise, S is said to be a non-linear code.

Theorem 2.2 [33, 35]

A code S is linear iff it is a group.

Proof

Suppose S is linear. Then
$S = \{W: HW^t = 0 \text{ for all code word } W\}$
Let $W_1, W_2 \in S$. Then
$H(W_1 + W_2)^t = H(W_1^t + W_2^t)$
$= HW_1^t + HW_2^t$
$= 0$
i.e. $W_1 + W_2 \in S$ and so S is a groupoid
Now,
$H(W_1 + (W_2 + W_3))^t = H((W_1 + W_2) + W_3)^t$
$= H(W_1 + W_2 + W_3)^t$
$\therefore$ S is a semigroup.
Also, if e is the identity element, then
$He^t = 0$
and so, S is a monoid.
Finally, given $W \in S$, $W^{-1} = W \in S$, where W^{-1} is the inverse of the word W.
Thus, S is a group.
Conversely, suppose S is a group. Then $e \in S$.
I.e. $W + W = e \in S$ for all W.
$H(e^t) = H(W + W)^t$
$= HW^t + HW^t$
$= 0 \in S$
i.e. $HW^t = 0$ for all $W \in S$
and so S is linear. QED □

3. METHODOLOGY

In this section, a general model for all categories of languages is first formulated in terms of programming languages. Thereafter, the data structure of the character set of FORTRAN programming language (modeled as Language I i.e. L1) is then used as an application of the model such that L1 is a subset of the ASCII (Language II i.e. L2), EBCDIC (Language III viz. L3) and standard BCD (Language IV i.e. L4) coding systems. A basic general result based on the model is then stated. The mathematical principles used are the concepts of sets, code presentation and groups, as explained in Section II.

3.1 **Formulation of the Computational Model as an Algorithmic Procedure in Terms of Programming Languages**

The general problem that has been tackled in this paper may be formulated as follows: Suppose a Language I (L1) is given whose basic character set is known but nothing is known about its meaning (i.e. nothing is known about the word which interprets the usage of every character). Now suppose there are some arbitrary languages, say Language II (L2), Language III (L3) and Language IV (L4) which belong to the same language family as L1, but unlike L1, the meanings are known. The character set of each of L2, L3 and L4 is a superset of that of L1. The task is to develop a model for tracing the meaning of elements of the character set of L1. The ultimate goal is to infer the characteristics of L1 within L2, L3 and L4. An algorithmic process or procedure for the formulation is presented below and depicted in Figure 3.1.

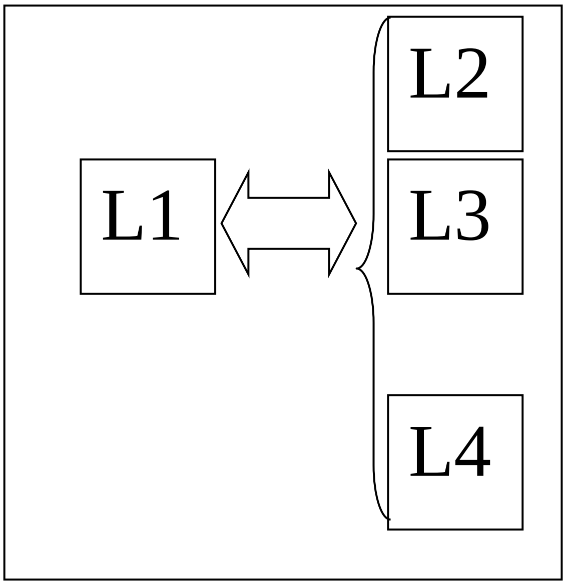

Figure 3.1: Schematic of Model Interaction between a Hypothetical Language (L1) and Three other Languages (L2, L3 and L4)

By simple analogy, L1, L2, L3 and L4 respectively correspond to Programming Language I (PrL1), Programming Language II (PrL2), Programming Language III (PrL3) and Programming Language IV (PrL4) in the procedure.

Procedure Language Characteristics

0. Start
1. Consider a Programming Language I (PrL1) whose character set is a subset of each of Programming Language II (PrL2), Programming Language III (PrL3) and Programming Language IV (PrL4).
2. Transform the character set of PrL1 into one which has meaning by using exactly the same meaning of the exact character in each of PrL2, PrL3 and PrL4.
3. By virtue of the inherent partition arrangement in the code presentation lossless data compression algorithm, use the algorithm to simplify the structure of the set of meanings of PrL1 as a subset of PrL2, PrL3 and PrL4.
4. Suppose MII* is the set of meanings of PrL1 as a subset of PrL2. Similarly, let MIII* be the set of meanings of PrL1 as a subset of PrL3 and MIV*is the set of meanings of PrL1 as a subset of PrL4. Then compare the characteristics of PrL2, PrL3 and PrL4 with set of meanings MII*, MIII* and MIV*.
5. Stop

3.2 **Result of the Model**

In this subsection, the code presentation algorithm in Section II is applied to the structure of FORTRAN character set (Γ) in the 8-bit ASCII, 9-bit EBCDIC and 9-bit standard BCD coding systems. In the presentation to follow, the subscripts AA, BB and CC respectively represent the structure of Γ in each of the coding systems. A basic characteristic of the coded character set (CCS) of FORTRAN programming language is that it consists of only the numerics, alphabetics as well as operation and special characters (OSC), and doesn’t contain control characters. It is a subset of the ASCII, EBCDIC and the Standard BCD code and so it is a subset of the Unicode Standard. There are several versions of FORTRAN such as FORTRAN 66, FORTRAN 4, FORTRAN 77, FORTRAN 90, FORTRAN 95, FORTRAN 2003, FORTRAN 2008, FORTRAN 2010, FORTRAN 2015 and Pro FORTRAN 2016, some of which include the concept of object-orientation [43]. The cardinality of the basic character set of FORTRAN varies with its version. In this paper, FORTRAN programming language with cardinality 48, in which the cardinality of its OSC is 12, is used [44]. The principle espoused in the paper is however applicable to all versions of the language, irrespective of the cardinality, including those whose cardinality is 49 [45] and those with cardinality 60 [46].

Definition 3.1

Let $\alpha = \{A, B, \ldots, Z\}$ and $\theta = \{0, 1, 2, \ldots, 9\}$.

Then the FORTRAN character set, Γ, is given by $\Gamma = \alpha \cup \theta \cup \beta$, where β = {+ - * / . ,’ = $ (   ) blank} is the set of special characters of Γ i.e. $/\beta/ = 12$.

<u>Theorem 3.1</u>

The zoned codes (z_i) and the decinumer sets (Q_i) of the numerics (θ), the upper case alphabetics (α) and the special characters (β) of Γ in the ASCII, EBCDIC and the standard BCD systems are respectively given by Table 2, based on the Type I definition of zoned portion:

Table 2: Zoned Codes and the Decinumer Sets of the FORTRAN Character Set in the ASCII, EBCDIC and Standard BCD Coded Character Sets

	z_i	**Q_i**
(i)		
θ_{AA}	0110	K_1
	0111	K_2
$\theta_{BB} = \theta_{CC}$	11110	K_1
	11111	K_2
(ii) α_{AA}	1000	K_3
	1001	K_1
	1010 1011	K_3 K_4
$\alpha_{BB} = \alpha_{CC}$	11000	K_5
	11001	K_{16}
	11010	K_7
	11011	K_6
	11100	K_{15}
	11101	K_6
(iii) β_{AA}	0100	K_8
	0101	K_9
β_{BB}	01000	K_{17}
	01001	K_{10}
	01011	K_{11}
	01100	K_6
	01101	K_{18}
	01111	K_{12}
β_{CC}	01000	K_{17}
	01001	K_{13}
	01010	K_{19}
	01011	K_{14}
	01100	K_6
	01101	K_{14}
	01111	K_{13}

where subscripts AA, BB, CC denote ASCII, EBCDIC and Standard BCD systems respectively, and Kj (j = 1, 2, …, 19) is given by Table 3:

Table 3: Table of Parameters K_j (j = 1, 2, …, 19)

j	K_j	$O(K_j)$
1	{1, 2, 4, 7, 8, 11, 13, 14}	8
2	{0, 3, 5, 6, 9, 10, 12, 15}	8
3	K_2 – {0}	7
4	{1, 2, 4}	3
5	K_1 – {1}	7
6	{1, 2}	2
7	K_2 – {3}	7
8	{0, 9, 15}	3
9	$K_1 \cup$ {15}	9
10	{7, 11, 13}	3
11	{1, 6, 10}	3
12	{11, 13}	2
13	{7, 8}	2
14	{6, 9}	2
15	K_7 – {0}	6
16	{0, 3}	2
17	{ 0 }	1
18	{ 6 }	1
19	{ 1 }	1

Proof

This follows from the combined application of Definition 2.7 and Definition 2.9. Essentially, the zoned code of the numerics of ASCII is the set {0110, 0111} of order 2 and with decinumer sets K_1 and K_2 each of order 8 corresponding to the two zoned portions 0110 and 0111 respectively. The numerics of both EBCDIC and Standard BCD systems behave in exactly the same way. That is, the zoned code of the numerics of each is the set {11110, 11111} of order 2 with decinumer sets K_1 and K_2 corresponding to the two zoned portions 11110 and 11111 respectively. As per the upper case alphabetics of ASCII, the zoned code is the set {1000, 1001, 1010, 1011} of order 4 with decinumer sets K_3, K_1, K_3 and K_4 respectively. Continuing this way, zoned codes and decinumer sets are obtained for the uppercase alphabetics of the EBCDIC and Standard BCD systems. Similarly, zoned codes and decinumer sets are obtained for the special characters of

ASCII, EBCDIC and Standard BCD systems as shown in the tables above. Hence the result. QED □

Theorem 3.2

The coded character set, Γ, of the FORTRAN programming language (PrL1) is a non-linear code in the 8-bit ASCII (PrL2), 9-bit EBCDIC (PrL3) and 9-bit Standard BCD (PrL4) coding systems.

Proof

The proof follows from the combined application of Definition 2.7, Definition 2.9, Equation 2.2 and Theorem 2.2. It shall be shown that Γ is a non-groupoid code in each of the three systems.

Let Z_{AA}, Z_{BB}, Z_{CC}, respectively represent the zoned codes of Γ in the ASCII, EBCDIC and Standard BCD systems.

Then,

Z_{AA} = {0110, 0111, 1000, 1001, 1010, 1011, 0100, 0101}

Z_{BB} = {11110, 11111, 11000, 11001, 11010, 11011, 11100, 11101, 01000, 01001, 01011, 01100, 01101, 01111}

Z_{CC} = (11110, 11111, 11000, 11001, 11010, 11011, 11100, 11101, 01000, 01001, 01010, 01011, 01100, 01101, 01111}

Now for all $W_1, W_2 \in Z_{AA}$, $W_1 + W_2 = e \notin Z_{AA}$ where e = 0000 is the identity element. Similar argument also applies to Z_{BB} and Z_{CC}. Hence the result. QED □

Theorem 3.3

Suppose a hypothetical Language I (L1), Language II (L2), Language III (L3) and Language IV (L4) correspond to PrL1, PrL2, PrL3 and PrL4 respectively as defined in Theorem 3.2. Then at least one of the following holds in HL1:

(i) The composition or concatenation of any two meanings or words of the language will not result into a valid meaning in the language.

(ii) The reverse meaning of the language will not necessarily be found in the language.

(iii) The identity element will not necessarily be present in the language.

(iv) The composition of any three meanings of the language will not necessarily result into the same meaning if the pairing of the three meanings is not the same.

Proof

By Theorem 2.2, since PrL1 is nonlinear in PrLI2, PrL3 and PrL4, then it is not a group code. This means elements of PrL1 don’t satisfy the group axioms. Hence the result. Alternatively, Theorem 2.1 (Lagrange’s Group Theorem) may simply be invoked to conclude that PrL1 is not a group code since the order of the character set of PrL1 is 48 and does not divide the order of PrL2, PrL3 and PrL4 which are respectively 256, 512 and 512. QED □

4. Discussion

Based on the code presentation algorithm used in this paper, the numerics of the coded character set of FORTRAN programming language, Γ, has two decinumer sets in each of the 8-bit ASCII (represented as AA), 9-bit EBCDIC (BB) and 9-bit Standard BCD (CC) coding systems. The uppercase alphabetics of Γ has four decinumer sets in AA and six decinumer sets each in BB and CC. Also the special characters of Γ has two decinumer sets in AA, six decinumer sets in BB and seven decinumer sets in CC, as reflected in Theorem 3.1.

An important task in showing the relationship between language L1 and the three other languages L2, L3 and L4 is to consider the order of L1 relative to each of L2, L3 and L4. If the order of L1 (i.e. o(L1)) does not divide the order of Li (where i = 2, 3, 4), then it can easily be concluded that L1 is not a group code in Li. However, when o(L1) divides o(Li), specific check still needs to be carried out to see if all the four axioms of a group (Definition 2.11) are satisfied. If the four axioms are satisfied, then it can be conjectured that L1 and Li belong to the same language family.

The mathematical concepts of a groupoid, monoid, semigroup and group provide a platform for ascertaining whether the composition or concatenation of elements of a language will produce same or similar elements in another language.

The procedure presented in this paper can be generally applied to the character set of all programming languages including (modern) object-oriented languages like Java, Visual Basic, C++ etc.

5. CONCLUSION

In this paper, a computational model for studying the characteristics of languages has been presented. That is, a general framework is presented for studying languages using the behavior of the FORTRAN character set in three coding systems. These systems are the 8-bit ASCII, 9-bit EBCDIC and 9-bit Standard BCD systems in which a parity check bit has been appended to each. The platform used for the model is code presentation, a technique for representing uniform binary digital computer codes. This technique happens to be a data compression algorithm which is lossless.

The use of code presentation in this paper has been primarily as a representation scheme, with minimal emphasis on the compression properties of the coded character sets. These properties have been substantially investigated and reported in other papers such as [19, 20].

Further work can be done to establish a mathematical condition to show when two languages belong to the same family using code presentation. For languages which originated from the same source and belong to the same family, the meanings of words in the evolving languages may be traced with respect to the source.

Also, the analysis done in this paper may be extended to the character sets of other programming languages such as Java, C/C++.

REFERENCES

[1]*Holy Bible*, King James Version, Evangel Publishers Ltd, 2007.

[2] Enfield J. (2017). 'Huh? Is That a Universal Word?', *American Scientist*, Volume 107, pp. 178-183.

[3] Niesler T., Louw P., and Roux J. (2005). 'Phonetic Analysis of Afrikaans, English, Xhosa and Zulu using South African Speech Databases', *Southern African Linguistics and Applied Language Studies*, 23 (4), 459-474.

[4] Harbert W. (2007). *The Germanic Languages* (Cambridge Language Surveys), Cambridge University Press.

[5] Roberge P. T. (2002). *Afrikaans – Considering Origins in Language in South Africa*, Cambridge University Press.

[6] Lewis M. P. (ed.) (2009), *Ethnologue: Languages of the World*, SIL International, P. Afrikaans; http://www.ethnologue.com/show_language.asp?code=afr

[7] https://en.wikipedia.org <last accessed in January 2020>

[8] Kuku A. O. (1980). *Abstract Algebra*. Ibadan University Press, Ibadan.

[9] Ilori S. A. and Akinyele O. (1986). *Elementary Abstract and Linear Algebra*, Ibadan University Press.

[10] Sebesta R. W. (1996). Concepts of Programming Languages, 3rd edn., Reading, Mass: Addison-Wesley Publishing Company.

[11] Bannister B. R. and Whitehead D. G. (1987). *Fundamentals of Modern Digital Systems.* Macmillan Education Ltd., Hampshire, Great Britain.

[12] Benedicty M. and Sledge F. R. (1987). *Discrete Mathematical Structures.* Harcourt Brace Jovanovich Inc, Orlando Florida.

[13] Berztiss A. T. (1975). *Data Structures-Theory and Practice.* Academic Press Inc., New York.

[14] Gorn S. (ed) (1966). Proposed American Standard Character Structure and Character Parity Sense for Parallel-by-Bit Data Communication in ASCII *Communications of the ACM* 9 (9), 695-697.

[15] Meadows R. and Parsons A. J. (1983). *Microprocessors: Essentials, Components and Systems.* Pitman Publishing, London.

[16] Strubble G. W. (1975). *Assembler Language Programming: The IBM System/ 360 and 370*. Addison-Wesley Publishing Company, Philippines.

[17] Oluwade B. (1998). Applications of 2-Code Error Detection Techniques, *Proceedings of the 14th National Conference* of COAN (Nigeria Computer Society), Vol. 9, 245-251.

[18] Oluwade B. (2004). Design and Analysis of Computer-Coded Character Sets, PhD Thesis, Department of Computer Science, University of Ibadan.

[19] Oluwade B. (2009). A Binary Text Compression Algorithm Based on Partitioning, *Advances in Computer Science and Engineering*, Vol. 3, Issue 2, 165 -174.

[20] Oluwade B. (2010). Application of a Data Compression Technique to the American Standard Code for Information Interchange (ASCII), *International Journal of Information Science and Computer Mathematics,* Vol.1, No.1, 1-7.

[21] Conder M., Havas G., Newman M. F. and Ramsay C. (2020). 'On Presentations for Unitary Groups', *Journal of Algebra*, Volume 545, pp. 100-110.

[22] Magnus W., Karrass A., and Solitar D. (1976). *Combinatorial Group Theory: Presentations of Groups in terms of Generators and Relations.* Dover Publications Inc., New York.

[23] Oluwade B. (2004). “Algebraic Characteristics of the CCITT#2 Communication Code,” *International Journal of Applied Mathematics & Statistics*, (IJAMAS), Vol. 2, No. M04, 50-59.

[24] Oluwade B. (2000). ‘A Novel Groupoid Code’, *Journal of Science Research* (Journal of the Faculty of Science, University of Ibadan, Nigeria), 6[1], 1-3.

[25] Oluwade B. (2011). ‘Design of a Byte-based Coded Character Set’, *The Journal of Computer Science and Its Application* (An International Journal of the Nigeria Computer Society), Vol. 18, No. 2, 42-54.

[26] Oluwade B. (2018). ‘Development of an Extended Text Compression Nomenclature and its Application to the Gray Codes in the Encoding of Chromosomes’, *The Journal of Computer Science & Its Applications* (Journal of the Nigeria Computer Society), Vol. 25, No. 2, pp. 115-130.

[27] Pujolle G., Seret D., Dromard D. and Horlait E. (1988). *Integrated Digital Communications Networks*, Vol. 1, John Wiley and Sons, New York.

[28] Ziv J. (2009). ‘The Universal LZ77 Compression Algorithm is Essentially Optimal for Individual Finite-Length N-Blocks’, *IEEE Transactions on Information Theory*, Vol. 55, Issue 5, pp. 1941-1944.

[29] Yoshida S., Morihara T., Yhagi H. and Saloh N. (1999). ‘Application of a Word Based Text Compression Method to Japanese and Chinese Texts’, *Proceedings of Data Compression Conference* (DCC’99), pp. 561.

[30] Osman M. Y., and Ai-Habib M. (1991). ‘Arabic Text Compression using Huffman Code’, *Arabian Journal for Science and Engineering*, Vol. 16, No. 4B, pp. 613-618.

[31] Mackenzie C. E. (1980). *Coded Character Sets, History and Development (The Systems Programming Series)*, Addison-Wesley Publishing Company, Inc, Philippines.

[32] Herstein I. (1975). *Topics in Algebra.* John Wiley & Sons, New York, 2nd edn.

[33] Slepian D. (1956). A Class of Binary Signalling Alphabets. *Bell System Technical J*ournal 35, 203-234.

[34] Oluwade B., Uwadia C.O. and Ayeni J. O. A. (2001). Asymptotic Time Complexity of an Algorithm for finding the Error Pattern of a Uniform Digital Code, *Journal of Scientific Research and Development* (Journal of the Faculty of Science, University of Lagos) , Vol.6, 127-134.

[35] Berlekamp E. R. (1968). *Algebraic Coding Theory.* McGraw-Hill Book Company, New York.

[36] Sun Y., Cui, C., Lu J. and Wang Q. (2016). ‘Data Compression and Reconstruction of Smart Grid Customers based on Compressed Sensing Theory’, *International Journal of Electrical Power & Energy Systems*, Vol. 83, 21-25.

[37] Mukhopadhyay S. K., Mitra S. and M. Mitra M. (2013). ‘ECG Signal Compression using ASCII Character Encoding and Transmission via SMS’, *Biomedical Signal Processing and Control*, Vol. 8, Issue 4, 354-363.

[38] Mukhopadhyay, S. K., Ahmad M. O, and Swamy M. N. S. (2017). ‘ASCII-Character-Encoding based PPG Compression for Tele-monitoring System’, *Biomedical Signal Processing and Control*, 31, 470-482.

[39] Suzuki T., Hasegawa S., Hamamoto T., and Aizawa A. (2011). ‘Document Recommendation using Data Compression’, *Procedia – Social and Behavioral Sciences*, 27, 150-159.

[40] Malik A., Sikka G. and Verma H. K. (2017). ‘A High Capacity Text Steganography Scheme based on LZW Compression and Color Coding’, *Engineering Science and Technology, an International Journal*, 20, Issue 1, 72-79.

[41] Yang C. and Chen J. (2017). ‘Chapter 4 – Efficient Nonlinear Regression-based Compression of Big Sensing Data on Cloud’, *Big Data Analytics for Sensor-Network Collected Intelligence* (A Volume in Intelligent Data-centric Systems), Academic Press, 83-98.

[42] Oluwade B. (2017). ‘Modelling of Components of Building Structures Using Discrete Structures of Computer Science’, *University of Ibadan Journal of Science and Logics in ICT Research* (Journal of the Department of Computer Science, University of Ibadan, Nigeria), Vol. 1, 25-33.

[43] www.fortran.com <last accessed July 2017>

[44] Lipschutz S. and Poe A. (1982). *Schaum’s Outline of Theory and Problems of Programming with Fortran*. Singapore: McGraw-Hill.

[45] www.fortan.com/F77_std/rjcnf-3.html <last retrieved July 2017>

[46] Metcalf M., Reid J. and Cohen M. (2011). *Modern Fortran Explained*, Oxford University Press.

[47] Penland A. and Sunic Z. (2019). ‘A Language Hierarchy and Kitchens-type Theorem for Self-Similar Groups’, *Journal of Algebra*, Volume 537, pp. 173-196.