LIMITATIONS IN THE CONVECTIONAL ISBN-10 CODE

Waweru Kamaku[1]

Cecilia Mwathi[2], **Bernard Kivunge**[3]

Jomo Kenyatta University of Agriculture and Technology[1,2]
Kenyatta University[3] (Kenya)

Abstract

The International Standard Book Number system (ISBN-10) which was in operation until 2007 uniquely identified every book published internationally but was later replaced with the ISBN-13. The code was had the ability to detect and correct single errors, to detect and correct some transposition errors and also detect multiple errors. This paper discusses some major limitations in the code and shows how error detection and correction capabilities affected the total dictionary on the code. The paper then shows the great need that lead to the development of the ISBN -13.

Keywords: Code, Dictionary, ISBN-10, Error detection, Error Correction.

INTRODUCTION

In this paper, the term dictionary shall be used to mean the total number of code words that can be generated by a code. In a code C, a symbol or digit, referred to as the check digit which helps to ensure that the code satisfies a certain condition may be added to in order to achieve some degree of redundancy. During communication, if $a = a_1\, a_2 \dots a_{10}$ and $b = b_1\, b_2 \dots b_{10}$ are the sent and received codeword respectively, situations may occur and you find that $a \neq b$, this simply mean that error(s) occurred. The ability to detect the existence of an error(s) in a code word is called error detection whereas error correction is the ability to correct the existing error(s) in a code word. Modular Arithmetic involves working with the remainders generated by division. If *a, b, c, d* be intergers, *a* is said to be congruent to *b* modulo *c* written $a \equiv b \text{ mod } (c)$ if $c\,|(a\text{-}b)$. That is c "divides" the difference (a-b). In other words, the number *d* is found such that (a-b) = cd.

In the late 1960's, book publishers realized that they needed a uniform way to identify all the different books that were being published throughout the world. In 1966, Gordon Foster, W. H. Smith and others they came up with the **I**nternational **S**tandard **B**ook **N**umber system (ISBN) which was later published by the international organization for standardization 1970 as international standard ISO 2108. Every book, including new editions of older books, was to be given a special number, called an ISBN, which is not given to any other book. According to Eric Weisstein, the ISBN-10 code consist of ten digits code words made up of any of all 10-digit decimal numbers 0, 1, 2, …9 and *X* for 10. Suppose $\boldsymbol{a} = a_1a_2a_{3\dots\dots}a_{10}$ is an ISBN-10 codeword, it must satisfy the condition $\sum_{i=1}^{10} ia_i \equiv 0 \ (mod\ 11)$,…I, called the parity-check equation. An ISBN is broken into groups from left to right which are the Prefix element, Registration group element, Registrant element, Publication element and the Check digit. In the event that the check digit is 10, the symbol *X* is used in the final position.

If $\boldsymbol{a} = a_1, a_2, \dots, a_{10}$ and $\boldsymbol{b} = b_1, b_2, \dots, b_{10}$ are the sent and received codeword respectively, situations may occur where $\boldsymbol{a} \neq \boldsymbol{b}$, this simply imply that error(s) occurred. A single error may occur as a result of incorrect typing of one digit in an ISBN codeword or due as a result of a smudge. A double error can occur when two digits in a codeword are incorrect. A silent error occurs when the parity check equation holds despite a mistyping in the codeword. A transpose error may occur due to interchanging of digits in a codeword this may lead to parity check equation not being met. The code uses calculation modulo 11 since 11 is a prime number and since it has no factors thus only multiples of 11 would yield to 0 mod 11. The strength of a code may be measured by its ability to detect and correct errors. This means that the encoding and decoding process should be as error free as possible to achieve this goal. A code that does not guarantee this (has many loop holes), even if it has a multi trillion dictionary, may render the code very weak to be depended on. The ISBN code is designed to detect single error and also double error which comes as a result of transposition of two digits.

Raymond, showed that to correct a single error the sent codeword, $\boldsymbol{a} = (a_1, a_2, \ldots, a_{10})$, has to satisfy the following: $\sum_{i=1}^{10} a_i \equiv \sum_{i=1}^{10} ia_i \equiv 0 \ (mod\ 11)$ ……..II. Similarly, to correct a transpose error the sent codeword $\boldsymbol{a} = (a_1, a_2, \ldots, a_{10})$ has to satisfy the following: $\sum_{i=1}^{10} a_i \equiv \sum_{i=1}^{10} ia_i \equiv \sum_{i=1}^{10} i^2 a_i \equiv \sum_{i=1}^{10} i^3 a_i \equiv 0 \pmod{11}$. (Raymond, 1986) ……..…..III

Proposition 1: Let $\boldsymbol{a}$ be an ISBN-10 code word. Then $\boldsymbol{a}$ does not necessarily satisfy all the conditions given equation II and III above.

Proof: (By counter example): Consider the code word **a** = 0198538030.

$\sum_{i=1}^{10} ia_i = 1*0 + 2*1 + 3*9 + 4*8 + 5*5 + 6*3 + 7*8 + 8*0 + 9*3 + 10*0 \equiv 0 \pmod{11}$. Therefore **a** = 0198538030 is an ISBN-10 code word. But $\sum_{i=1}^{10} a_i \equiv 4 \pmod{11} \neq 0 \pmod{11}$. Since one of the conditions is not satisfied, then $\boldsymbol{a}$ does not necessarily satisfy all the conditions. This result shows that not every ISBN-10 code word will satisfy all the conditions. Thus if the conditions are imposed so that the code can correct errors, then some code words will be excluded.

Proposition 2: The ISBN-10 code has a dictionary with an upper limit of 10^9 = 1,000,000,000 code words. Since the digit repetition selection for the first nine positions is allowed, there are 10^9 permutations. Thus if no other conditions are attached to the code (for example error detection and correction), it has a dictionary of 10^9 = 1,000,000,000 code words. If digit repetition was not allowed, there would have been ${}_{10}P_9$ = 3,628,800 code words.

Proposition 3: In an ISBN-10 code, the check bit string does not permute.

Proof: Computation for the check digit is done modulo 11.

Let C_1 and C_2 be two ISBN 10 code words given by $C_1 = a_1\, a_2 \ldots a_9\, a_{10}$ and $C_2 = a_1\, a_2 \ldots a_9\, a'_{10}$. Since C_1 and C_2 are a code words, then $(\sum_{i=1}^{9} ia_i) + 10\, a_{10} \equiv 0 \pmod{11}$ and $(\sum_{i=1}^{9} ia_i) + 10\, a'_{10} \equiv 0 \pmod{11}$. But C_1 and C_2 are similar in the first nine bit strings thus $10a_{10} = 10a'_{10}$. By cancellation laws for integers. $a_{10} = a'_{10}$. Hence the check bit string does not permute. #

Proposition 4: The ability to detect an error(s) in the ISBN-10 code does not affect (increase or reduce) its dictionary size.

Proof: The parity check equation is the general condition for a code word to be an ISBN-10 code word. Thus to detect an error(s), there is no extra condition imposed. With this initial condition, the code will still generate the expected code words which satisfy the condition. Any arbitrary code word which does not satisfy this condition will not be an ISBN-10 codeword thus does not increase or reduce the dictionary. If a silent error occurs, it yields a different codeword which is also a member of the code and similarly does not increase or reduce the dictionary.

Proposition 5: The conditions for error correction given by equation VI and VIII above reduces the size of the dictionary size in the ISBN-10.

Proof. (By counter example): Consider the code word C_1 = 0198538030. $(\sum_{i=1}^{10} ia_i) \equiv 0 \pmod{11}$. Therefore C_1 is an ISBN-10 code word. But $(\sum_{i=1}^{10} a_i) \equiv 4 \pmod{11} \neq 0 \pmod{11}$. Then $\boldsymbol{a}$ does not necessarily satisfy the equation VI above. Thus if the code is to correct errors, C_1 will not be contained in it since it does not satisfy the required conditions. This in turn reduces the number of code words in the code. #

Proposition 6: The ability to correct a single error in the ISBN-10 code does not affect (increase or reduce) its dictionary size.

Proof : $\sum_{i=1}^{10} a_i = a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9 + a_{10} \equiv 0 \pmod{11}$.

$\sum_{i=1}^{10} ia_i = a_1 + 2a_2 + 3a_3 + 4a_4 + 5a_5 + 6a_6 + 7a_7 + 8a_8 + 9a_9 + 10a_{10} \equiv 0 \pmod{11}$

Thus $a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9 + a_{10} = 11k,\ k \in \mathbb{Z}$

$a_1 + 2a_2 + 3a_3 + 4a_4 + 5a_5 + 6a_6 + 7a_7 + 8a_8 + 9a_9 + 10a_{10} = 11r,\ r \in \mathbb{Z}$

Equation ii - i yields $a_2 + 2a_3 + 3a_4 + 4a_5 + 5a_6 + 6a_7 + 7a_8 + 8a_9 + 9a_{10} = 11(k - r)$, where $(k - r) \in \mathbb{Z}$ which means that $a_2 + 2a_3 + 3a_4 + 4a_5 + 5a_6 + 6a_7 + 7a_8 + 8a_9 + 9a_{10} \equiv 0 \pmod{11}$.

This equation does not involve a_1 and hence a_1 can be chosen from 0 to 9. Since the check digit a_{10} does not permute, the other bit strings $a_2, a_3, \ldots, a_8$ can be chosen from 0 – 9 such that equation XIV holds. Solving this Diophantine equation requires that the GCD of 2, 3, 4, 5, 6, 7, 8 and 9 divide 11(k - r). The GCD is 1 which will always divide 11(k - r) and thus the equation have integer solution.

The bit strings $a_2, a_3, \ldots, a_8$ can be chosen from 0 to 9 yielding to 10^7 choices. In total we have 10^8 choices which show that the dictionary is not affected.

Proposition 7: The ability to correct transpose errors in the ISBN-10 code does not affect (increase or reduce) its dictionary size. The proof of this can be shown and follows the same procedure as Proposition 6.

Conclusion and Recommendations.

In ISBN-10, Silent errors can go unnoticed, to detect an error in ISBN-10 code word, one has to work out the parity check equation which makes the process tedious unless the error is an omission or insertion error which obviously can be detected by just counting the number of digits to confirm if they are less or more than required. ISBN-10 can only correct single errors and transpose errors and for the transpose errors there has to be prior knowledge of the existence of the transpose error. The process for correcting the errors as show above may end up being tedious and only applicable to code words which meet some criteria. These conditions given in equation I I and III only tell if one can correct the double error and do not show how to do correct it. Multiple(more than two) errors cannot be corrected. ISBN10 has a relatively small dictionary. The great demands for a bigger dictionary lead to the design of ISBN-13 code which is now in use since year 2007. There is need further research to show if the improved ISBN-13 code dictionary size affected the error detection and error correction capability.

References

1. Eric Weisstein., (2010). ISBN Code http://mathworld.wolfram.com/Transposition.html. Date accessed: 5th May 2010
2. Eric Weisstein., (2010). ISBN Code. http://mathworld.wolfram.com/ISBN.html. Date accessed: 5th May 2010
3. Kenneth H. Rosen, (1993). *Elementary number theory and its applications.* Addison-Wesley Publishing company.
4. Leo E., (2005). The Coding of the ISBN. http://en.scientificcommons.org/leo_egghe. Date accessed: 6th May 2010
5. Nyaga L. and Cecilia M., (2008). Increasing error detection and correction Efficiency in the ISBN. *Discovery and Innovation* **20:** 3 and 4.
6. Raymond H., (1986). *A first course in coding theory*. Oxford University Press.
7. Ronald R. and Leo E., (2005). On the detection of double errors in ISBN and ISSN-like codes. http://en.scientificcommons. org/leo_egghe. Date accessed: 6th May 2010
8. Viklund A., (2007). ISBN Information Home. http://isbn-information.com/index.html Date accessed: 5th May 2010.