Investigation 35

Check Digits

Focus Questions

By the end of this investigation, you should be able to give precise and thorough answers to the questions listed below. You may want to keep these questions in mind to focus your thoughts as you complete the investigation.

- What are check digits, and how are they used?
- What are some common check digit schemes for credit card and ISBN numbers?
- What is Verhoeff's check digit scheme, and how is it related to the dihedral group of order 10?

Preview Activity 35.1. This investigation will involve both congruence of integers and the dihedral group D_5. In this activity, we will review a few of the basics.

(a) Determine the value of x that satisfies the congruence equation

$$2(1) + 3(2) + 4(2) + 5(6) + 6x \equiv 0 \pmod{7}.$$

(b) Consider the permutation $\pi = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 5 & 7 & 6 & 2 & 8 & 3 & 0 & 9 & 4 \end{pmatrix}$ in S_{10}. (Note that we are permuting the digits 0 through 9 instead of 1 through 10.)

(i) Explain why π is an even permutation.

(ii) What is the order of π?

(iii) Find $\pi^4(5)$, where π^4 denotes the composition of 4 copies of π (that is, $\pi \circ \pi \circ \pi \circ \pi$).

(iv) If x is any integer between 0 and 9, what is $\pi^8(x)$? Explain.

Introduction

In this investigation, we will discuss check digits and explore various check digit schemes, including one that uses the dihedral group D_5. Check digits are important because nowadays most information is transmitted electronically. When we use an ATM or pay with a credit card, there is always

the possibility that an error in transmission can occur. For instance, noise can be introduced in a message, information can be lost, and data can be altered during transmission. Another important source of error is human error. Data can be confused when humans enter numbers into machines or communicate to others. Some mistakes are more prevalent than others. For example, according to Richard Hamming,* the two most common human errors when dealing with information are interchanging digits (e.g., typing 12 instead of 21) and changing one of a string of three digits when two adjacent digits are the same (e.g., using 112 instead of 122). Clearly, problems can arise when data is not encoded or transmitted properly. Fortunately, there are ways to compensate for these errors. The first step is to determine when they occur, and that is where check digits come into play.

Check Digits

A check digit is a digit appended to a string, usually at the end, to make the sum of the digits in that string congruent to a specific number modulo a given integer. For example, the 10 digit identification number 2361068754 might have an extra digit d appended to the end so that the digit sum is congruent to 0 modulo 9. In this case, the check digit would be 3 and the ID number would be 23610687543. As their name suggests, check digits perform a check to ensure that the number received is a valid number. However, just because the number appears to be valid, that does not necessarily make it legitimate. For instance, a credit card company may choose to only use a small subset of the set of valid credit card numbers. We should also note that even when check digits are used to detect errors, they do not necessarily provide a way to correct the errors they find. (There are other methods for doing that.)

Check digits are used in UPC codes, credit card numbers, ISBN numbers, and most other identification systems. We will now consider a few of the more common and/or mathematically interesting check digit schemes.

Credit Card Check Digits

Different credit cards have account numbers, or ID codes, of different lengths and with different prefixes. Each code consists of a string of numbers, with each digit between 0 and 9. The prefix of a card is the one or two digit block at the beginning (the leftmost digits) of the ID code. In particular:

- MasterCard codes have 16 digits and use prefixes of 51, 52, 53, 54, and 55.
- VISA codes have either 13 or 16 digits and use a prefix of 4.
- American Express codes have 15 digits and use prefixes 34 or 37.
- Discover codes have 16 digits and use a prefix of 6011.

The prefix is the Major Industry Identifier (MII) and indicates the type of industry that issues the card. For example, VISA and MasterCard are issued by the banking and financial sector (with prefix

**Coding and Information Theory* (2nd ed.), Prentice-Hall, 1986, p. 27.

numbers 4 or 5) while American Express is in the travel and entertainment category (prefix number 3). All credit cards compute a check digit modulo 10.[†] To find the check digit, we can use a process known as the *Luhn algorithm*:[‡]

(1) Beginning with the second digit from the right (in other words, don't include the check digit) and moving from right to left, double every other digit. Add the individual *digits* of these numbers (e.g., if the doubled number is 16, add 1 and 6).

(2) Sum the digits (but not the check digit) not considered in step (1).

(3) Add the results of steps (1) and (2). Call this result s.

(4) The check digit d is the solution to $s + d \equiv 0 \pmod{10}$.

Activity 35.2.

(a) Find the correct check digit d for the sample VISA card with number 4417 1234 5678 911d.

(b) Create your own valid American Express card number.

ISBN Check Digits

The acronym ISBN is an abbreviation for the International Standard Book Number, which is used to identify books. An ISBN-10 has the form

$$X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10},$$

where each X_i is a digit between 0 and 9. In an ISBN, the first digit, X_1, represents the language of the book (0 is English), the next block (2 or 3 digits) identifies the publisher, the third block (5 or 6 digits) is a publisher's number for the book, and the last digit is a check digit.

The check digit in an ISBN is determined by first attaching a weight to each digit, with the leftmost digit (X_1) having a weight of 1, the next digit (X_2) having a weight of 2, and so on. (In general, the weight of the digit X_k is k.) Next, we multiply each digit (except the check digit) by its weight and compute the weighted sum. The check digit is congruent to the weighted sum modulo 11, with X representing a check digit of 10.

A quick way to implement this scheme is through the use of weight vectors and dot products. Recall that the dot product of vectors $[v_1, v_2, \ldots, v_n]$ and $[w_1, w_2, \ldots, w_n]$ is the scalar

$$[w_1, w_2, \ldots, w_n] \cdot [v_1, v_2, \ldots, v_n] = \sum_{i=1}^{n} w_i v_i.$$

By taking the dot product of the first 9 digits of an ISBN with the weight vector $[1, 2, 3, 4, 5, 6, 7, 8, 9]$, we can easily determine what the check digit should be.

Activity 35.3. The ISBN-10 for a very interesting book is 0-471-33193-?. Find the check digit.

[†]Some books and web sites state that MasterCard and VISA use the prefix digits when determining the check digit, while American Express and Discover do not. To the best of our knowledge, all companies use the prefix digits in their calculations, and so we will do the same.

[‡]The Luhn Algorithm was created by Hans Peter Luhn, who worked for IBM. It is patented in U.S. Patent No. 2.950.048.

This ISBN-10 check digit scheme will find all single digit errors, but will also catch errors obtained by interchanging digits (for example, typing 12 instead of 21). However, the ISBN-10 scheme is restricted to ID numbers with 10 digits, and we have to introduce the extra symbol X to represent the digit 10 as a possible check digit. In the next section, we will examine a check digit scheme that works for ID numbers with any number of digits.

Verhoeff's Dihedral Group D_5 Check

In the late 1960s, Dutch mathematician Jacobus Verhoeff proposed a check digit scheme based on the dihedral group D_5.[§] This scheme is an improvement on others in that it works for any length number, and it detects all single digit errors and all transposition errors involving two adjacent digits. However, the Verhoeff scheme is a little more complicated to implement.

We begin with the operation table for D_5 given in Table 35.1. (Note that the elements in this table are listed in a different order than usual; this is done to match Verhoeff's labeling.)

	I	R	R^2	R^3	R^4	rR^4	rR^3	rR^2	rR	r
I	I	R	R^2	R^3	R^4	rR^4	rR^3	rR^2	rR	r
R	R	R^2	R^3	R^4	I	rR^3	rR^2	rR	r	rR^4
R^2	R^2	R^3	R^4	I	R	rR^2	rR	r	rR^4	rR^3
R^3	R^3	R^4	I	R	R^2	rR	r	rR^4	rR^3	rR^2
R^4	R^4	I	R	R^2	R^3	r	rR^4	rR^3	rR^2	rR
rR^4	rR^4	r	rR	rR^2	rR^3	I	R^4	R^3	R^2	R
rR^3	rR^3	rR^4	r	rR	rR^2	R	I	R^4	R^3	R^2
rR^2	rR^2	rR^3	rR^4	r	rR	R^2	R	I	R^4	R^3
rR	rR	rR^2	rR^3	rR^4	r	R^3	R^2	R	I	R^4
r	r	rR	rR^2	rR^3	rR^4	R^4	R^3	R^2	R	I

Table 35.1
Operation table for D_5.

We then replace the elements in the D_5 table with the digits 0 to 9 (keeping the elements in the same order as in Table 35.1) to obtain Table 35.2:

Verhoeff's check digit scheme requires an ID number of the form $a_{n-1}a_{n-2}\cdots a_1a_0$ (note that the digits are indexed from right to left, starting with an index of 0) to satisfy the equation

$$\pi^0(a_0)\cdot\pi^1(a_1)\cdot\pi^2(a_2)\cdots\pi^{n-1}(a_{n-1}) = 0,$$

where

$$\pi = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 5 & 7 & 6 & 2 & 8 & 3 & 0 & 9 & 4 \end{pmatrix}$$

[§] J. Verhoeff, "Error detecting decimal codes," *Mathematical Centre* Tract 29, The Mathematical Centre, Amsterdam, 1969.

$\cdot$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

Table 35.2
Operation table for the Verhoeff check digit scheme.

is a permutation, $\pi^i = \pi \circ \pi \circ \cdots \circ \pi$ is the composition of i copies of π, and the $\cdot$ operation is that which arises from D_5 as indicated in Table 35.2. Note that the permutation applied to each digit depends on the position of the digit in the ID number. For example, if 4 is the digit in the third position from the right, then we apply π^2 to 4 to obtain 7. The result of applying the powers of π to any position can be described in the *permutation table* shown in Table 35.3, where the i^{th} row shows the result of applying the permutation π^i to each possible digit. Note that the powers of π are periodic, so the rows repeat after row 8. In other words, $\pi^{i+8}(k) = \pi^i(k)$ for all k.

	0	1	2	3	4	5	6	7	8	9
π^0	0	1	2	3	4	5	6	7	8	9
π^1	1	5	7	6	2	8	3	0	9	4
π^2	5	8	0	3	7	9	6	1	4	2
π^3	8	9	1	6	0	4	3	5	2	7
π^4	9	4	5	3	1	2	6	8	7	0
π^5	4	2	8	6	5	7	3	9	0	1
π^6	2	7	9	3	8	0	6	4	1	5
π^7	7	0	4	6	9	1	3	2	5	8

Table 35.3
Permutation table for the Verhoeff check digit scheme.

Now that we understand some of the mechanics involved, the Verhoeff scheme can be implemented as follows:

(1) Start with the n digit number

$$a_{n-1}a_{n-2}a_{n-3}\cdots a_1a_0,$$

with the digits labeled from right to left, starting with a_0.

(2) Let c denote the *checksum*, and set $c = 0$ initially.

(3) Step through the n-digit number digit by digit, each time replacing c with $c \cdot \pi^i(a_i)$ (where the operation $\cdot$ is indicated in Table 35.2).

The original number has a valid check digit if and only if, at the end of the process, the checksum c is equal to 0.

As an example, Table 35.4 shows how the steps described above can be followed to validate the checksum for the ID number 4134705. Since the final value of c is 0, we have a valid ID number.

i	a_i	$\pi^i(a_i)$	Old c	New c: Old $c \cdot \pi^i(a_i)$
0	5	5	0	5
1	0	1	5	9
2	7	1	9	8
3	4	0	8	8
4	3	3	8	5
5	1	2	5	8
6	4	8	8	0

Table 35.4
A Verhoeff check digit example.

The Verhoeff algorithm detects all single digit errors and all transposition errors made by interchanging two adjacent digits. Also, the Verhoeff algorithm detects over 95% of twin errors (where aa is changed to bb), over 94% of jump transpositions (abc replaced with cba) and jump twin errors (aca replaced with bcb), and most phonetic errors ($a0$ replaced with $1a$—for example, 40 replaced with 14; notice how these two numbers sound similar when read aloud). Gallian[¶] describes an interesting application in which the German government used a slight modification of the Verhoeff scheme to append check digits to serial numbers on their banknotes.

Activity 35.4. How do we find the check digit for an ID number using the Verhoeff scheme? Consider the ID number $1023857d$, where d is the check digit.

(a) Determine the value of the checksum c for the related ID number 10238570.

(b) Let $d = c^{-1}$. Show that $1023857d$ is a valid ID number. (As it turns out, this technique will always yield a correct check digit. Exercise (5) asks you to prove this result.)

Concluding Activities

Activity 35.5. Since 2007, books have been identified with 13 digit ISBNs (the ISBN-13). The first 12 digits of the ISBN-13 contain the identifying code and the 13th digit is the check digit. The

[¶] *Contemporary Abstract Algebra* (5th ed.), Houghton Mifflin Company, 2002.

check digit scheme is similar to the one used in ISBN-10 IDs. Research the ISBN-13 check digit scheme and explain how it works. Be sure to cite all of your sources in your explanation. Then find the check digit d for the ISBN-13 978-082183798d.

Exercises

(1) Determine if each of the following is a valid ID number. Assume that the last digit of each number is the check digit. If the check digit is not correct, fix it.

(a) 10513485, using the Luhn algorithm

(b) The ISBN-10 ID number 0-131-87718-4

(c) 2401346782, using the Verhoeff scheme.

(2) **Airline ticket ID numbers.** Airlines tickets have a 15-digit identification number. The first digit (reading left to right) is the coupon number and identifies the leg of the trip. (Coupon number 1 indicates the first flight in the trip, 2 the second, and so on, while coupon number 0 is the customer's receipt.) The next three digits identify the airline, and the next 10 digits comprise the document number, while the last digit is the check digit. Airlines use a simple mod 7 system to determine the check digit. If the airline ticket has digits

$$d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}d_{11}d_{12}d_{13}d_{14}d_{15},$$

then the check digit d_{15} satisfies

$$d_{15} \equiv d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}d_{11}d_{12}d_{13}d_{14} \pmod 7.$$

(a) Verify that 1-101-2134601379-2 is a valid airline ticket. Use a computer algebra system if necessary.

(b) Calculating the remainder when dividing a 14-digit number by 7 is easy for a computer algebra system but a bit time-consuming for humans. Here we will investigate one method that can be done by hand: the method of *casting out sevens*. This method works by determining the remainders when dividing powers of 10 by 7.

(i) Determine a general formula for calculating $10^n \pmod 7$ for nonnegative integers n.

(ii) Expand 11012134601379 in powers of 10, and then use your answer to part (a) to calculate the remainder when 11012134601379 is divided by 7.

(c) Another algorithm that can be used to determine if a large number is divisible by 7 is the following:

- Remove the last (rightmost) digit from the number.
- Subtract twice the value of the removed digit from the remaining number.
- Repeat until you can tell if the number obtained is divisible by 7.

(i) Apply this algorithm to show that 11012134601377 is divisible by 7. Then explain why the algorithm works.

(ii) Can a method similar to this one, where we remove the last digit and subtract some single-digit multiple of that digit from the new number, be used to test for divisibility by any other single digit integer? If so, find and explain all of the cases in which such a method can be used.

(3) In this exercise, we will examine the types of errors that are detected (or not detected) by the Luhn algorithm.

(a) Does the Luhn algorithm find all single digit errors? That is, if

$$a_{n-1} \cdots a_{i+1} a_i a_{i-1} \cdots a_0 d$$

and

$$a_{n-1} \cdots a_{i+1} b_i a_{i-1} \cdots a_0 d'$$

are valid IDs with $a_i \neq b_i$ and check digits d and d', must d and d' be different? If the answer is yes, prove it. If no, provide a counterexample.

(b) Will the Luhn algorithm detect all errors obtained by interchanging digits (e.g., typing 12 instead of 21)? If the answer is yes, prove it. If no, provide a counterexample.

(4) In this exercise, we will examine the types of errors that are detected (or not detected) by the ISBN-10 scheme.

(a) Show that the ISBN-10 scheme detects all single digit errors. (See part (a) of Exercise (3).)

(b) Show that the ISBN-10 scheme detects all errors obtained by interchanging digits (for example, typing 12 instead of 21).

(c) Another common family of errors are twin errors. One example of a twin error is changing aa to bb. Does the ISBN-10 scheme detect all twin errors? If the answer is yes, prove it. If no, provide a counterexample.

(d) Two other types of common errors are jump transposition errors (for example, abc replaced with cba) and jump twin errors (for example, aca replaced with bcb). Does the ISBN-10 scheme detect these errors? If the answer is yes, prove it. If no, provide a counterexample.

* (5) Activity 35.4 provided a method for finding a correct check digit using the Verhoeff scheme. Prove that this method works in general. That is, show that if the original ID number ends in 0 and has checksum c, then c^{-1} (from Table 35.2) is the correct value to use as the check digit in place of the final 0.

(6) In this exercise, we will examine the types of errors that are detected (or not detected) by the Verhoeff scheme.

(a) Show that the Verhoeff scheme detects all single digit errors. (See part (a) of Exercise (3).)

(b) Complete the following steps to show that the Verhoeff scheme detects all errors obtained by interchanging digits (for example, typing 12 instead of 21).

(i) Show by direct calculation that if $a \neq b$ in D_5, then $a \cdot \pi(b) \neq \pi(a) \cdot b$.

(ii) Extend the result of the previous part to show that if $a \neq b$ in D_5, then $\pi^{i-1}(a)\pi^i(b) \neq \pi^i(a)\pi^{i-1}(b)$ for all $i \in \mathbb{Z}^+$.

(iii) Now show that the Verhoeff scheme detects all errors obtained by interchanging digits.

Connections

Abstract algebra has many practical applications, and this investigation considered one of these applications. In particular, we saw how congruence of integers (from Investigation 2) and a dihedral group (from Investigation 21) can be used to create check digit schemes. These check digit schemes help make transfers of information more reliable and are therefore important components of our electronic world.