A New Class of Check-Digit Methods for Arbitrary Number Systems

H. PETER GUMM

Abstract—**For arbitrary number systems we present a new check-digit method that detects all single-digit errors and all transpositions of adjacent digits using a single check digit from the given number system. In previous methods at least one type of transpositional error had to remain undetected. The key to this method lies in using the dihedral groups together with appropriate transformations in the important cases, where the numbers are represented in base $2r$ with r odd.**

I. Introduction

Empirical studies have shown [9], [10] that the most common typing errors that occur when data are entered on a keyboard are

- single-digit errors (one digit wrong)
- format errors (one digit inserted or left out)
- transpositions (interchanging of two adjacent digits).

To detect such errors, the original string of data is supplied with one check digit, where digit now means a "digit" in the chosen number system, i.e., the check digit is numerical for numerical data and may be alphanumeric for alphanumeric data. Various check-digit methods have been designed for the decimal number system. Each method is able to detect all "single-error mistakes," but they fail to detect all transposition errors. At least one (and often not more than one) erroneous transposition is undetectable with those methods, see [1]–[3].[1]

The format errors principally cannot all be detected by only one check digit, and every method that detects single-digit errors will automatically detect about 90 percent of all format errors. The use of two check digits as proposed in [3] and [11] is not advised since the previously mentioned studies have also shown that the absolute number of errors that occur roughly doubles when the number of digits increases by two. Thus generally specifying and checking for a fixed format seems appropriate to detect format errors. We therefore concentrate on methods to detect single-digit errors and transposition errors.

Clearly, for numbers to the base 2 such a check digit method is impossible since the numbers 00, 01, 10 would have to be supplied with mutually different check digits from the set $\{0,1\}$.

If a number n has the digits $d_k, d_{k-1}, \cdots, d_1$ in base r, i.e., $n = \sum d_i r^{i-1}$, then using $p = -\sum d_i$ (in modulo r arithmetic) for a check digit will detect every single-digit error. The secured number then has the digit representation $d_k, d_{k-1}, \cdots, d_1, p$, and

Manuscript received August 26, 1983; revised January 16, 1984.

The author is with the Gesellschaft für Strahlen und Umweltforschung München–Neuherberg, Institut für Medizinische Informatik und Systemforschung, D–8042 Oberschleissheim, Federal Republic of Germany.

[1]After this paper was written, I was informed that J. Verhoeff in his thesis "Error correcting decimal codes," (*Mathematical Centre Tracts*, no. 29, Amsterdam: Mathematisch Centrum, 1969) has presented a different check-digit method. Our method, nevertheless, appears to be more general and simpler from a conceptual as well as an implementational standpoint.

0018-9448/85/0100-0102$01.00

to check for correctness, we check whether the digits add to zero (modulo r). Since addition (modulo r) is commutative, a transposition error $\cdots ab \cdots \rightarrow \cdots ba \cdots$ will never be detected. Therefore, conventional methods impose a weight on every digit position before adding, or, more generally, use permutations σ_i of the base digits $\{0,1,\cdots,r-1\}$ on every digit position [1]–[3]. Thus the check digit is computed as $p = -\sum\sigma_i(d_i)$ (modulo r). Again single-digit errors will be detected for any chosen sequence σ_i of permutations, but the particular choice of the sequence $(\sigma_1, \sigma_2, \cdots)$ will determine the properties in detecting transpositions.

If r is prime, then it is easy to see that choosing numbers $a_i \in \{1,\cdots,r-1\}$ with $a_i \neq a_{i+1}$, and defining $\sigma_i(x) \triangleq a_i x$ (modulo r) will detect all transposition errors. Let us call this method a "weighted parity check method".

The method fails when r is even; e.g., in the case $r = 10$. First, only those numbers a that are relatively prime to 10 may be used for weights, to guarantee single-error detection. As a consequence, $(a_i - a_{i-1})$ will be even, so $(a_i - a_{i-1})$ is a zero-divisor in the ring of integers modulo 10, which implies that those transpositions $\cdots d_i d_{i-1} \cdots \rightarrow \cdots d_{i-1} d_i \cdots$ will not be detected where $(d_i - d_{i-1}) \equiv 5$ (modulo 10).

Choosing different kinds of permutations σ_i will not solve the dilemma. To detect the transposition $\cdots d_i d_{i-1} \cdots \rightarrow \cdots d_{i-1} d_i \cdots$, the σ_i, σ_{i-1} will have to satisfy

$$\sigma_i(d_i) + \sigma_{i-1}(d_{i-1}) \neq \sigma_i(d_{i-1}) + \sigma_{i-1}(d_i) \pmod{10}$$

for every pair $d_i \neq d_{i-1}$, $d_i, d_{i-1} \in \{0,\cdots,9\}$. This means that $\tau(x) \triangleq \sigma_i(x) - \sigma_{i-1}(x)$ must be a permutation of the set $\{0,\cdots,9\}$. In [7] we have shown that for any two permutations α,β on the numbers $\{0,\cdots,9\}$ the map $\alpha - \beta$ (modulo 10) can never be permutation again. The proof is valid for every number system with base $r = 2k$.

Some conventional methods pretend that the unsecured decimal numbers were written in base 11 and use a check-digit method in base 11, but then the methods have to take nonnumerical "digits" into account to represent the 10th digit in base 11. The International Standard Book Number (ISBN) classification of books, for example, uses the letter "X" as a last "digit" in such cases.

II. General Results

According to the previous discussion, we arrive at the following abstract definition.

Definition: A check-digit method base r is a set D with $|D| = r$ together with a family $F \triangleq (f_1, f_2, \cdots, f_n, \cdots)$ of operations f_n: $D^n \rightarrow D$ such that the following axioms are satisfied for all $n \in \mathbb{N}$:

$$f_n(x_n,\cdots,x_i,\cdots,x_1) = f_n(x_n,\cdots,x_i',\cdots,x_1) \Rightarrow x_i = x_i' \tag{1}$$

$$f_n(x_n,\cdots,x_i,x_{i-1},\cdots,x_1) = f_n(x_n,\cdots,x_{i-1},x_i,\cdots,x_1) \Rightarrow x_i = x_{i-1} \tag{2}$$

$$f_n(x_n,\cdots,x_2,f_n(x_n,\cdots,x_1)) = x_1 \Rightarrow f_n(x_n,\cdots,x_1) = x_1. \tag{3}$$

Clearly, $f_k(d_k,\cdots,d_1)$ is to be interpreted as the check digit for the number that has the digit representation $d_k,\cdots,d_1$ in base r. Without loss of generality, we may assume that $D = \{0,1,\cdots, r-1\}$. Clearly, axiom (1) guarantees single-error detection, while (2) and (3) guarantee detection of transposition errors. Note that (3) is needed to detect transposition of the check digit with its left neighboring digit. The above definition specifies a universal algebra $A = (D, F)$. (We refer the interested reader to the monographs of Cohn [4] or Grätzer [6].) Since the axioms are universally quantified implications, standard universal algebraic results tell us that they are inherited in passing to direct products. The following proposition presents this idea more concretely.

Proposition: If there exist check-digit methods base r and base s, then there is a check-digit method base rs.

Simply represent every digit base rs uniquely as a pair (d_1, d_2), where d_1 is a digit base r and d_2 a digit base s. Then compute check digits p_1 and p_2 separately for both components in their respective bases and reconvert the pair (p_1, p_2) into the corresponding number base rs. It is easily checked that the properties (1)–(3) are preserved. Since there are check-digit methods available for base q, where q is an odd prime, the Proposition yields check-digit methods for every odd number base. Note that there is also a different way to obtain check-digit methods for base r when r is a prime power, including the case that $r = 2^m$ with $m > 1$. If $r = p^s$, then choose, as before, weights $(a_1, a_2, \cdots, a_n \cdots)$ with $a_i \in \{1,\cdots,r-1\}$ and $a_i \neq a_{i+1}$. If the number n is expressed as $d_k, d_{k-1},\cdots,d_1$ in base r, then compute the check digit for n as $f_k(d_k, d_{k-1},\cdots,d_1) \triangleq -\sum a_i d_i$, where multiplication and addition now are performed in the Galois field $GF(p^s)$. The properties of a check-digit method follow from the field properties just like in the standard weighted parity check methods.

Again we see why the case $p^s = 2$ is excluded: there is no way to choose the sequence $(a_1, a_2, \cdots)$ with the required properties in GF(2). Nevertheless, the important cases of hexidecimal-numbers ($r = 16$) and alphanumeric data ($r = 36 = 4*9$) are thus covered, yet others like numerical ($r = 10$) or alpha-strings ($r = 26$) still are not captured.

III. The Case $r = 2s$ With Odd s

In view of the results of the previous section, we still need a check-digit method for numbers base $r = 2s$, where s is an odd prime. Here we give a method for $r = 2s$, where s is any odd number greater than or equal to three. A weighted parity check based on addition modulo r is impossible, as was argued in Section I. Thus, we want to replace addition by a different kind of operation, which we shall call $*$. Clearly, $*$ will have to be cancellative at both sides, i.e., $a * x = a * y$ implies $x = y$ and $x * a = y * a$ implies $x = y$ for any $a, x, y \in \{0,1,\cdots,r-1\}$. Thus $*$ must be a quasi-group operation [5], and to be able to discard brackets when "adding" more than two numbers, we demand associativity, i.e., $a*(x*y) = (a*x)*y$. It follows that $*$ must be a group operation, since it acts on the finite set $\{0,1,\cdots,r-1\}$. In case $r = 2p$, with p an odd prime, there are precisely two groups on r elements, namely the cyclic group $\boldsymbol{Z}_r$ and the dihedral group $\boldsymbol{D}_p$ [8].

Thus again let s be an odd number greater than 2 and $r = 2s$. The dihedral group $\boldsymbol{D}_s$ of order $r = 2s$ may be represented as the set of all pairs (e, x) with $e \in \{-1, 1\}$ and $x \in \{0,1,\cdots,s-1\}$, where $*$ is defined as

$$(e,x)*(f,y) \triangleq (ef, ey + x)$$

and the inner multiplications and addition are evaluated in the cyclic group $\boldsymbol{Z}_s$ of order s. This notation is shorthand for the matrix representation

$$\begin{pmatrix} e & x \\ 0 & 1 \end{pmatrix}$$

of the elements of D_s with $*$ being matrix multiplication. For $a, b \in \boldsymbol{Z}_s$ with $a \neq 0$ let us define a map τ: $D_s \rightarrow D_s$ by

$$\tau(e,x) = (e, e(a - x) + b).$$

Then we have the following lemma.

Lemma: τ is a permutation on D and satisfies for all $u, v \in D_s$: $\tau(u) * v = \tau(v) * u$ implies $u = v$.

Proof: If $u = (e, x)$ and $v = (f, y)$, then $\tau(u) = \tau(v)$ implies $e = f$ and $e(a - x) + b = f(a - y) + b$. Thus $x = y$, so τ

is a permutation. Suppose that $\tau(u) * v = \tau(v) * u$, then with the same notation

$$(e, e(a-x)+b) * (f, y) = (f, f(a-y)+b) * (e, x)$$

so

$$(ef, ey + e(a-x) + b) = (ef, fx + f(a-y) + b).$$

Hence

$$e(a+y-x) = f(a+x-y)$$

i.e., $(e-f)a = (e+f)(x-y)$. If $e \neq f$, then the right-hand side becomes zero, so we get $2a = 0$, which is impossible since s is odd and multiplication takes place in the cyclic group $\boldsymbol{Z}_s$. Thus $e = f$, which again implies $2(x-y) = 0$; hence $x = y$.

Now we may define a check-digit method for $r = 2s$, using τ and its iterates $\tau, \tau^2, \tau^3 \cdots$. (Here $[\cdot]^{-1}$ denotes the inverse in the dihedral group.

Theorem 1: With any τ as in the Lemma the definition

$$f_n(x_n, \cdots, x_1) \triangleq \left[\tau^n(x_n) * \tau^{n-1}(x_{n-1}) * \cdots * \tau(x_1)\right]^{-1}$$

yields a check-digit method base $r = 2s$.

Proof: Verifying axiom (2) is easy.

$$\begin{aligned}\Big[\tau^n(x_n) * \cdots * \tau^{i+1}(x_{i+1}) * \tau^i(x_i) * \tau^{i-1}(x_{i-1}) \\ * \tau^{i-2}(x_{i-2}) * \cdots * \tau(x_1)\Big]^{-1} \\ = \Big[\tau^n(x_n) * \cdots * \tau^{i+1}(x_{i+1}) * \tau^i(x_{i-1}) * \tau^{i-1}(x_i) \\ * \tau^{i-2}(x_{i-2}) * \cdots * \tau(x_1)\Big]^{-1}\end{aligned}$$

yields after taking inverses and cancellation of equal terms

$$\tau^i(x_i) * \tau^{i-1}(x_{i-1}) = \tau^i(x_{i-1}) * \tau^{i-1}(x_i).$$

Thus

$$\tau\big(\tau^{i-1}(x_i)\big) * \tau^{i-1}(x_{i-1}) = \tau\big(\tau^{i-1}(x_{i-1})\big) * \tau^{i-1}(x_i).$$

By using the Lemma we get

$$\tau^{i-1}(x_i) = \tau^{i-1}(x_{i-1})$$

and since τ is one-to-one, this results in $x_i = x_{i-1}$.

Axiom (3) requires a few more calculations. Assuming that $f_n(x_n, \cdots, x_2, f_n(x_n, \cdots, x_1)) = x_1$, i.e.,

$$\Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau\Big(\Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\Big]^{-1}\Big)\Big]^{-1} = x_1$$

we infer, using the group laws

$$\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau\Big(\Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\Big]^{-1}\Big) = x_1^{-1}$$

$$\tau\Big(\Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\Big]^{-1}\Big) = \Big[\tau^n(x_n) * \cdots * \tau^2(x_2)\Big]^{-1} * x_1^{-1}.$$

Multiplying with $\tau(x_1)^{-1}$ from the left, we get

$$\tau(x_1)^{-1} * \tau\Big(\Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\Big]^{-1}\Big) = \Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\Big]^{-1} * x_1^{-1}$$

$$\tau(x_1)^{-1} * \tau\Big(\Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\Big]^{-1}\Big) * x_1 = \Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\Big]^{-1}$$

$$\tau\Big(\Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\Big]^{-1}\Big) * x_1 = \tau(x_1) * \Big[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\Big]^{-1}.$$

Now the conclusion of the Lemma can be used to yield

$$\left[\tau^n(x_n) * \cdots * \tau^2(x_2) * \tau(x_1)\right]^{-1} = x_1,$$

i.e. $f_n(x_n, \cdots, x_1) = x_1$

Combining with the Proposition we obtain the following theorem.

Theorem 2: For every number $r \neq 2$ there exists a check digit method for the base r number system.

IV. Modifications

Given a number system with base r, where r is composite, there usually exist many different product decompositions of r to design different check-digit methods. Moreover, the method just given for $r = 2s$ with odd s allows modifications. For example, as a corollary to the proof of the Theorem we obtain the following.

Corollary: Let $r = 2s$ with s odd, and let $(\tau_1, \tau_2, \cdots, \tau_n, \cdots)$ be any sequence of permutations of $\{0, 1, \cdots, r-1\}$, all satisfying the conclusion of the Lemma. Define $\alpha_1 \triangleq \tau_1$ and $\alpha_i \triangleq \tau_i \circ \alpha_{i-1}$ for $i > 1$, then

$$f_n(x_n, x_{n-1}, \cdots, x_1) \triangleq \left[\alpha_n(x_n) * \alpha_{n-1}(x_{n-1}) * \cdots * \alpha_1(x_1)\right]^{-1}$$

yields a check-digit method for base r. Moreover, if the base r digit 0 is assigned to the pair $(1, 0)$ in the representation of the dihedral group $\boldsymbol{D}_s$ and if we choose $a = -b$ in the definition of τ, we get $\tau(0) = 0$, so leading zeros in the number to be encoded will have no effect on the check digit.

V. An Example

The following example shows that our method is easy to use and implement on a computer. Let $\boldsymbol{D}_5$ be the dihedral group of order 10, and let the elements $(1, x)$, resp. $(-1, x)$ be represented by the decimal digits x, resp. $5 + x$. Then Table I describes the image of the operation $*$ in the decimal digits. With the same correspondence the permutation τ: $\boldsymbol{D}_5 \to \boldsymbol{D}_5$, given by $\tau(e, x) \triangleq (e, e(1-x) - 1)$, corresponds to the permutation σ of the decimal digits, which is given by (14) (23) (58697) in cycle notation.

TABLE I
The Multiplication Table of $\boldsymbol{D}_5$

*	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

For example, the number 1793 will then obtain the check digit

$$\left[\sigma^4(1) * \sigma^3(7) * \sigma^2(9) * \sigma(3)\right]^{-1} = [1 * 6 * 5 * 2]^{-1} = [4]^{-1} = 1.$$

Thus the secured number will be 17931.

Checking a number for correctness is as easy as this method. The number 17931 is found to be correct, since $\sigma^4(1) * \sigma^3(7) * \sigma^2(9) * \sigma(3) * 1$ evaluates to zero. A simple transpositional error, 19731 say, would be detected, since $\sigma^4(1) * \sigma^3(9) * \sigma^2(7) * \sigma(3) * 1$ results in $4 \neq 0$.

References

[1] A. M. Andrew, "A variant of modulus 11 checking," *Comput. Bull.*, vol. 14, pp. 261–265, 1970.

[2] D. A. H. Brown, "Biquinary decimal error detection codes with one, two and three check digits," *Comput. J.*, vol. 17, pp. 201–204, 1974.

[3] C. K. Chu, "A note on multiple error detection in ASCII numeric data communication," *J. Ass. Comput. Mach.*, vol. 28, pp. 265–269, 1981.

[4] P. M. Cohn, *Universal Algebra*. Dordrecht, West Germany: D. Reidel, 1981.

[5] J. Denes and A. D. Keedwell, *Latin Squares and Their Applications*. New York: Academic, 1974.

[6] G. Grätzer, *Universal Algebra*, 2nd ed. New York: Springer, 1979.

[7] H. P. Gumm, "Encoding of numbers to detect typing errors," preprint, submitted.

[8] M. Hall, *The Theory of Groups*. New York: Macmillan, 1959.

[9] W. Kunerth, *Die EDV-gerechte Verschlüsselung: Grundlagen und Anwendungen moderner Nummernsysteme*. Stuttgart: Forkel, 1981.

[10] H. Schechinger, "Kontrolle mit Hilfe von Prüfziffern," *Bürotechnische Sammlung, BTS systematisch*, vol. 171, 1979.

[11] A. S. Sethel, V. Rajaraman, and P. S. Kenjale, "An error correcting scheme for alphanumeric data," *Inform. Process. Lett.*, vol. 7, pp. 72–77, 1978.

0018-9448/85/0100-0105$01.00