Note on Reinforcing Control Information in Variable Length Items

C. V. Ramamoorthy

Honeywell-E.D.P. Division, Needham, Massachusetts

INTRODUCTION

When an item in a data processing operation consists of a variable number of characters of fixed length, there arises the problem of indicating the end (or beginning) of such a string. There exist a number of ways to provide this “control” information; the most popular are the partitioned representation, the grid-matrix representation and the chained representation.

Since any error in the control information will make the identification of subsequent items either impossible or very expensive, a method of “reinforcing” the control information against errors is proposed. This paper considers the problem of coding special “partition symbols” which can serve as end markers for sequences of coded information symbols where the length of the sequences vary between strings.

The types of errors that can occur are unwarranted change of bits in the item (a “zero” changing to a “one” and vice-versa) and the addition or deletion of bits within an item.

From the standpoint of physical hardware, the addition or the deletion of bits are adequately protected by such schemes as special timing signals, clock tracks, tape gaps, etc. Thus, the first type of error (bit-interchange) is more common and there should be adequate protection against it. Such an error can mutilate the partition symbol so that it will be mistaken for an information character. Thus, two adjacent items will be erroneously read as one. Also, an error in some information character may make the latter a partition symbol in which case an item will be broken up into two. The following scheme is proposed to protect against such errors.
A METHOD OF REINFORCED PARTITION SYMBOLS

In an \( n \)-bit code structure there are \( 2^n \) code words. Choose a partition symbol from this set, and delete from this set all code words which are within a specified distance \( d \) from this code. Use the reduced list for code assignments of information characters. Thus, the partition character will be at least distance \( d \) from any information code word. However, no restriction is imposed on the distances between the information codes themselves. Thus, at least a \( d \)-bit error is needed either in the partition symbol or in an information symbol before any undetectable damage is done.

As an example, consider an alphanumeric 6-bit code with a distance-3 partition symbol so that a double error can be detected or a single error can be corrected if it occurred on the partition symbol.

We note that the code structure has 42 usable symbols. If 111 111 is used as the partition symbol, the code includes the common representation of the binary-coded-decimal code thus allowing all the usual flexibility of the arithmetic operations. Equation (1) gives the number of usable codes \( N \) when the number of bits per character \( n \) and minimum partition symbol distance \( d \) are specified and provided so that the number of partition symbols \( N_p \) is less than \( n + 2 \), which usually is the case.

\[
N = 2^n - \left\{ \sum_{i=0}^{d-1} \binom{n}{i} + \sum_{i=1}^{N_p-1} \binom{n - i}{d - i} \right\}.
\] (1)

It may seem that the problem of detection of the partition symbol and the problem of testing the information codes for validity is complex, but there is a simple technique which can accomplish this. We start by computing the distance between the partition symbols and each incoming code. The distance to an information code in the example should be 3 or greater for it to be a legal code. If the distance is 2, then the incoming code is probably an information code with a single error. If the distance is one, then it is probably a partition symbol with a single error. Thus, the decoding problem is simple.

To find all the usable code words in a specific case: Select all code words at distance 1 from each partition symbol by inverting their bits one at a time. List these and remove any duplicate code words. Now use these code words to generate distance-2 words by a similar procedure, again removing any duplicate words. Repeat this procedure until dis-
tance-\((d-1)\) codes from the partition symbols are found. These are all
the codes to be discarded from the possible \(2^n\) code words to get the
usable codes.

Let \(N\) be the number of usable information code words of \(n\) bits,
and let \(N_p\) be the number of partition symbols. Then the number of
useful code words is \(N + N_p\). We define the information content of a
code as \(\log_2{(N + N_p)}\) bits, the efficiency of the system as \(\eta = \frac{\log_2{(N + N_p)}}{n}\), and the redundancy of the system as \(1 - \eta\).

(a) If \(N_p = 1\), \(n = 6\) and \(d = 3\), then \(N + N_p = 43\),
\[
\eta = \log_2{(43/6)} = 91\%,
\]
and

\[
\text{redundancy} = 9\%.
\]
(b) Or, if \(N_p = 2\), \(n = 6\), \(d = 3\), then
\[
N + N_p = 34
\]
\[
\eta = 85\%,
\]
\[
\text{redundancy} = 15\%.
\]
The above computations indicate that our storage requirements will
be increased by about 9\% (for \(N_p = 1\)) from using all the \(2^n = 64\)
combinations with equal probability. In the practical case there may
not be any loss because we would still represent \(26 + 10 + 1 = 37\)
alphanumeric characters and space with 6-bit combinations and rarely
use the remaining combinations.

OTHER METHODS OF REINFORCEMENT

An alternative method of reinforcement of the partition symbol—
that by its repetition—can be expensive in storage. In some schemes
(Bemer, 1960) every \(n\)th character is examined for the partition symbol.
If it is not a partition symbol, the next \(n\)th character is examined. This
method requires the scanning of every \(n\)th character with the partition
symbol as the last character. The proposed scheme of partition symbol
reinforcement applies for error protection.

ANALYSIS OF THE REINFORCED PARTITION SYMBOL METHOD

We briefly give the main results of our analysis on the Reinforced
**NOTE ON REINFORCING CONTROL INFORMATION**

**TABLE I**

**TABLE OF NUMBERS OF USABLE INFORMATION CODES FOR CERTAIN VALUES OF $n$, $d$, $N_p$**

<table>
<thead>
<tr>
<th>$n$</th>
<th>$d$</th>
<th>$N_p$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>58</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>1</td>
<td>22</td>
</tr>
</tbody>
</table>

**THEOREM 1.** A partition symbol in a $n$-bit code has $\binom{n}{k}$ code words distance-$k$ ($0 \leq k \leq n$) from itself.

**LEMMA.** A code with a partition symbol distance $d$ away from the rest of the words has $\sum_{i=0}^{n} \binom{n}{i}$ useful code words.

If more than one partition symbol is needed at a minimum distance $d$ ($0 < d \leq n$), one has to specify also the relationship (distances) between the partition symbols themselves. The enumeration of usable words is quite complex. However, the number of partition symbols needed in a practical code is seldom greater than $n + 1$. In most cases, there will be only one partition symbol and sometimes 2.

**THEOREM 2.** If $N_p < n + 2$ is the number of partition symbols needed with minimum distance $d$ ($d < n$) from any legal information code word, then an optimum assignment (to give a maximum number of usable words) of partition symbols will have one partition symbol such that the other $(N_p - 1)$ partition symbols are exactly distance 1 away from it.

**THEOREM 3.** If $N_p < n + 2$ then the maximum number of useful code words distance-$d$ ($d < n$) away from all the partition symbols is given by

$$N = 2^n - \left\{ \sum_{i=0}^{d-1} \binom{n}{i} + \sum_{l=1}^{N_p-1} \binom{n-l}{d-1} \right\}.$$

**THEOREM 4.** For fixed values of $N_p$, and $d$

$$\left( \frac{N_{n+k}}{2^{n+k}} \right) > \left( \frac{N_n}{2^n} \right)$$

$n =$ length of code word

$k =$ any positive integer $> 0$

$N_n =$ No. of usable information code words of length $n$
i.e., the ratio of numbers of usable information codes to the total possible codes increases with \( n \).

Thus, for longer codes, the information efficiency increases.

EXTENSION

When error detection (or correction) is desired in information symbols, one can require that the distances between the information symbols be at least greater than a specified minimum. Thus, for \( n = 6 \), \( N_p = 1 \), \( d = 3 \), there would be 26 information code words which are at least distance-3 from each other. One can also extend this method so that the distances between the multiple partition symbols would be such as to permit error correction. This technique is applicable in all those areas of digital communication where there is a need for protection of control signals and symbols, whose mutilation will cause subsequent processing either too expensive or impossible. Thus, the scheme with appropriate modifications can be used to indicate the terminal end-markers for the information stored in digital delay lines and drums.

ACKNOWLEDGMENT

The writer is grateful to Prof. G. Salton and Mr. A. Anger of the Harvard Computation Laboratory for suggestions and encouragement, and to Dr. R. B. Lawrance and the Honeywell organization for subsequent support.

Received: January 25, 1963

REFERENCES
