A GROUP-THEORETIC FRAMEWORK FOR FAULT-TOLERANT COMPUTATION

Paul E. Beckmann

Massachusetts Institute of Technology
Research Laboratory of Electronics
Room 36-615
50 Vassar Street
Cambridge, MA 02139

Bruce R. Musicus

Bolt, Beranek, & Newman
Room 6/456
10 Moulton St.
Cambridge, MA 02138

ABSTRACT

In this paper we present a general technique for protecting computation with systematic-separate codes. These codes use parity symbols to check the result of computation. We use a group-theoretic approach and model computation as operations in an algebraic group. We show that in order for a code to commute with computation, it must define a homomorphism between the original group and the group of parity symbols. We then apply a quotient group isomorphism and reduce the problem of finding coding schemes to that of finding normal subgroups. In many instances, our method can be shown to identify all possible systematic-separate codes. For a given code, we present conditions on errors such that they may be detected and corrected. We briefly discuss the extension of our technique to other algebraic systems and conclude with two examples.

1. INTRODUCTION

Fault-tolerance is needed in many signal processing applications to ensure continuous operation and check the integrity of results. Traditionally, the problem of fault-tolerance has been solved by using modular redundancy. This is a general technique which can be applied to any system. However, it is computationally expensive and does not provide a high level of fault protection.

A more efficient method of protecting computation is to encode operands using an arithmetic code. This is an error-correcting code which commutes with arithmetic operations, i.e., its error detecting and correcting properties are preserved during computation. Arithmetic codes offer performance and redundancy advantages similar to existing error-correcting codes used to protect communication channels. Initial research in this area focused on protecting integer computation, and several useful codes resulted [1].

Recently, arithmetic codes have been extended to protect higher-level operations by encoding entire sequences of real or complex data. This field has been called Algorithm-Based Fault-Tolerance (ABFT) because redundancy is added to protect an entire algorithm, rather than individual arithmetic operations. The majority of ABFT schemes rely on linear error-correcting codes to protect linear operations. Examples of such operations include linear transformations [2], matrix operations [3], and A/D conversion [4]. Linear codes have also been used to protect convolution [5, 6]. Application of ABFT to other arithmetic operations has been limited by the absence of suitable coding schemes.

In this paper, we present a general method of developing systematic-separate arithmetic codes for protecting group operations. We use a group-theoretic approach similar to [7, 8] but generalize to encompass a wider range of operations. First, we present the basic structure of these codes and give a procedure for finding all possible codes. Next, we discuss the extension of our technique to other algebraic systems. Then, we present examples. A more detailed discussion of our technique is given in [9].

2. FRAMEWORK

In this section, we present a framework for analyzing fault-tolerant systems. We rely heavily upon the concepts and theorems of group theory, stating common results without proof. We refer the reader to [10] for appropriate mathematical background material.

A group $G = [\mathcal{G}; \square, 0_\square]$ is an algebraic system that consists of a set of elements, $\mathcal{G}$, a binary operation called the group product, $\square$, and an identity element, $0_\square$. Groups model a broad range of computation, including many useful signal processing operations, as well as ordinary integer and real number arithmetic.

The operation which we intend to protect is the group product $v = a \square b$ of two elements $a, b \in G$. We add fault-tolerance by adding redundant information which we exploit to detect and correct errors. In this paper we focus on a specific form of redundancy which yields *systematic-separate codes* (SSC). These codes are characterized by the use of parity symbols to check the result of computation [1].

The basic structure of a system computing $v = a \square b$ which is protected by an SSC is shown in Figure 1. We use this as a starting point in our analysis. Although we do not derive this structure here, it can be deduced from first principles by making only a few basic assumptions. An SSC contains the following basic steps:

Dr. Musicus is a Research Affiliate of the RLE Digital Signal Processing Group, M.I.T., and a Senior Scientist at Bolt, Beranek, & Newman. This research was funded in part by a Rockwell Doctoral Fellowship, in part by Draper Laboratory, Inc., under Grant No. DL-H-418472, and in part by the Advanced Research Projects Agency monitored by ONR under Grant No. N00014-89-J-1489.

0-7803-0532-9/92 $3.00

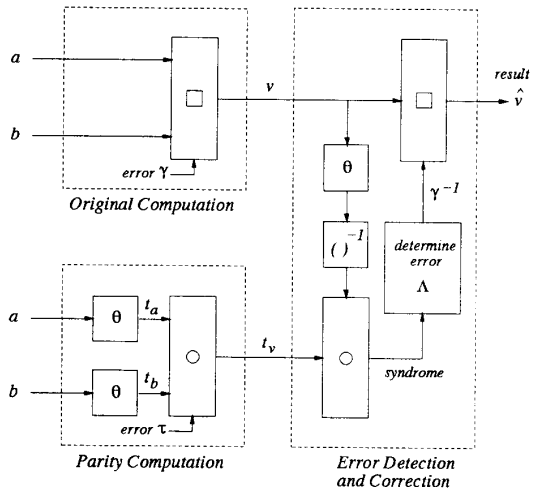

Figure 1: General form of a group operation protected by a systematic-separate arithmetic code.

1. Compute parity symbols $t_a = \theta(a)$ and $t_b = \theta(b)$ from the operands, where t_a and t_b are members of $T = [T; \mathsf{O}, 0_{\mathsf{O}}]$ a group of parity symbols, and θ is a mapping from G to T.
2. Compute the product of parity symbols, $t_v = t_a \mathsf{O} t_b$.
3. Compute the syndrome

$$S(v, t_v) = \theta(v)^{-1} \mathsf{O} t_v \tag{1}$$

and use it to detect and correct errors. If $S(v, t_v) = 0_{\mathsf{O}}$, we declare that no errors have occurred. Otherwise, if $S(v, t_v) \neq 0_{\mathsf{O}}$, we declare that some error has occurred.

It can be shown that the syndrome is a condensation of all information relevant to error detection and correction. Furthermore, in order to be able to correct all errors, each error must have a unique syndrome.

We begin by ignoring errors and determining conditions on θ such that $S(v, t_v) = 0_{\mathsf{O}}$ in an error-free system. Substituting into (1) and letting $S(v, t_v) = 0_{\mathsf{O}}$, we find that θ must satisfy

$$\theta(a \square b) = \theta(a) \mathsf{O} \theta(b) \tag{2}$$

for all $a, b \in G$. This equation is recognized as the defining property of an *algebraic homomorphism*. A homomorphism θ is a mapping from one algebraic system, G, to another, T, which preserves structure [10].

We now make an important simplifying assumption about θ. We assume that θ maps G *onto* T. This assumption is made for three reasons. First, requiring θ to be onto ensures that T will have the same number as, or fewer elements than, G. Hence, the complexity of the parity operation O will most likely be the same, or less than the complexity of the original product $\square$. Second, all elements of T will be used, yielding a greater level of fault-tolerance for a given parity channel complexity. Third, the structure of G will be heavily reflected in T, and we can use a well-known isomorphism involving quotient groups of G to identify possible groups T and homomorphisms θ.

The isomorphism is as follows. Let G be a group and let N be a normal subgroup of G. Denote by G/N the *quotient group* of G by N. The elements of G/N are cosets (sets of elements). The product in G/N is the product of subsets. Given G/N, we can always define a mapping ψ from G onto G/N by $\psi(g) = N \square g$. ψ maps g to the coset containing g. If we apply ψ to the product $a \square b$, we find that $\psi(a \square b) = \psi(a) \square \psi(b)$. Hence, ψ is a homomorphism from G onto G/N. Letting $\tilde{T} = [\tilde{T}; \tilde{\mathsf{O}}, \tilde{0}_{\mathsf{O}}] = G/N$ and $\tilde{\theta}(g) = N \square g$, we have a procedure for finding possible groups $\tilde{T}$ and homomorphisms $\tilde{\theta}$. By selecting different normal subgroups N, we obtain different possible parity groups $\tilde{T}$ and mappings $\tilde{\theta}$.

It can be shown that this procedure is guaranteed to find, up to isomorphisms, all possible groups $\tilde{T}$ and mappings $\tilde{\theta}$ which satisfy the assumed structure. Although we have not explicitly found all possible parity groups and homomorphisms, we have isomorphic copies of them, which have equivalent error detecting and correcting properties.

Note that when we implement a system which uses this isomorphism, we will *never* perform computation in the actual quotient group $\tilde{T}$ by manipulating cosets. Rather, by forming the quotient group, we will recognize that G/N is isomorphic to a simpler group T which we obtain by renaming the elements of G/N. This simpler group will be used for parity computations.

The problem of finding groups T and homomorphisms θ has been reduced to that of finding normal subgroups. For an arbitrary group, finding all normal subgroups is still a difficult problem. However, for many groups that compute useful arithmetic operations, finding normal subgroups is a trivial task. In these instances, we are able to determine all systematic-separate coding schemes.

We now include the possibility of errors occurring in our system, and for a given T and θ, show which types of errors may be detected and corrected. We use a product error model and denote by γ and τ the errors in $\square$ and O, respectively. The faulty results of the main and parity channels are assumed to be given by

$$\begin{aligned} v &= a \square b \square \gamma \\ t_v &= t_a \mathsf{O} t_b \mathsf{O} \tau. \end{aligned} \tag{3}$$

(We would obtain slightly different results if we assume that errors appear on the left hand side of the products.) Denote by $\mathcal{E}_\gamma$ the set of all possible errors γ and define $\mathcal{E}_\tau$ in a similar manner. These sets depend on the hardware architecture and specific computational steps used to compute the products.

In order to be detectable, all faulty results must yield nonzero syndromes. Substituting (3) into (1), we find that under errors γ and τ the syndrome equals

$$S(v, t_v) = \theta(\gamma)^{-1} \mathsf{O} \tau = S(\gamma, \tau). \tag{4}$$

Thus the syndrome is a function of the error only. All errors in the sets $\mathcal{E}_\gamma$ and $\mathcal{E}_\tau$ can be detected if and only if

$S(\gamma,\tau) \neq 0_{\bigcirc}$ for all $\gamma \in \mathcal{E}_\gamma, \tau \in \mathcal{E}_\tau$. To be correctable, errors must be detectable and each error $\gamma \in \mathcal{E}_\gamma$ must have a unique syndrome. Mathematically, this translates to $S(\gamma_i,\tau_i) \neq S(\gamma_j,\tau_j)$ for all $\gamma_i \neq \gamma_j \in \mathcal{E}_\gamma$, and $\tau_i, \tau_j \in \mathcal{E}_\tau$.

If the errors are correctable, then a simple, but not necessarily efficient, fault correction procedure could use the syndrome as an index into a lookup table Λ. Let $\Lambda(S(\gamma,\tau)) = \gamma^{-1}$. Since each error has a unique syndrome, this function is well-defined. Then, to correct the error and obtain the error-free result $\hat{v}$, apply $\Lambda(S(\gamma,\tau))$ to the result of the product in $\square$,

$$\hat{v} = v \square \Lambda(S(\gamma,\tau)). \tag{5}$$

It can be shown that this does indeed correct the error. Note that we do not correct the error τ in $\bigcirc$ since we are only interested in the result of the product in $\square$. The steps needed to perform error detection and correction are also illustrated in Figure 1.

Error correction via lookup tables is efficient only when the set of possible errors $\mathcal{E}_\gamma$ is small. In practice, systems exist in which $\mathcal{E}_\gamma$ is large or even infinite. In these instances, other techniques must be used. The usual approach taken is to exploit the structure of $\mathcal{E}_\gamma$ and to invert (4) over the set $\mathcal{E}_\gamma$.

If G is an Abelian (commutative) group, then a series of operations may be protected by performing error detection and correction only once at the end, rather than after each group product. Consider the result of N operations in an Abelian group G, and assume that errors occur during the $i^{\underline{th}}$ and $j^{\underline{th}}$ products in the main and parity channels, respectively. Then $v = g_1 \square \cdots \square g_{i+1} \square \gamma \square g_{i+2} \square \cdots \square g_{N+1}$ and $t_v = t_{g_1} \bigcirc \cdots \bigcirc t_{g_{j+1}} \bigcirc \tau \bigcirc t_{g_{j+2}} \bigcirc \cdots \bigcirc t_{g_{N+1}}$. Since G is commutative, T will be also. We can rearrange the products as follows: $v = (g_1 \square \cdots \square g_N) \square g_{N+1} \square \gamma$ and $t_v = (t_{g_1} \bigcirc \cdots \bigcirc t_{g_N}) \bigcirc t_{g_{N+1}} \bigcirc \tau$. This is equivalent to the original error model (3) and previous results still hold. The syndrome test is still applicable, and requirements for error detectability and correctability are unchanged. This yields more efficient systems since the overhead for fault-tolerance is distributed over several operations.

Now assume that errors are restricted to the system computing $\square$. (This is reasonable if the computational complexity of $\square$ is much greater than that of $\bigcirc$.) Then the syndrome of a faulty result equals $s = \theta(\gamma)^{-1}$. Let $K_\theta = \{g \in G \mid \theta(g) = 0_{\bigcirc}\}$. This is known as the *kernel* of the mapping θ and is similar to the nullspace of a linear transformation. It can be shown that $K_\theta = N$, where N is the normal subgroup used to define θ. Then all errors $\gamma \in \mathcal{E}_\gamma$ are detectable if and only if $\mathcal{E}_\gamma \cap K_\theta = \varnothing$. This makes intuitive sense since, if $\gamma \in K_\theta$, then $s = \theta(\gamma)^{-1} = 0_{\bigcirc}$, and hence γ is undetectable.

To determine a suitable coding scheme for a given operation, the following iterative procedure would be used:

1. Pick a normal subgroup N_i of G.
2. Determine the resulting parity group $\tilde{T}_i$ and mapping $\tilde{\theta}_i$ using the quotient group isomorphism. Then determine an isomorphic simpler group T_i and the corresponding mapping θ_i from G to T_i.
3. Determine a system for computing $\bigcirc$, and the sets of errors $\mathcal{E}_\gamma$ and $\mathcal{E}_\tau$ from the hardware architecture.
4. Check if the errors $\mathcal{E}_\gamma$ and $\mathcal{E}_\tau$ can be detected and corrected by θ_i. If they can, then we have found a suitable code. Otherwise, pick another normal subgroup and repeat from step 2.

Although this is an iterative procedure, it can still be quite fruitful. Also, normal subgroups usually have a standard form, so types of detectable and correctable errors can be determined easily.

3. Other Algebraic Systems

The concepts of homomorphism and quotient group occur in a wide variety of algebraic systems and can be used to define systematic-separate codes in a similar manner. We briefly state the extension of our framework to several other algebraic systems.

A ring $R = [\mathcal{R}; \square, \boxtimes, 0_\square]$ is a set $\mathcal{R}$ with two operations $\square$ and $\boxtimes$ which satisfy certain basic axioms [10]. Under $\square$, the elements of R form a group. A mapping θ from R to another ring $T = [\mathcal{T}; \bigcirc, \otimes, 0_\bigcirc]$ is a homomorphism if

$$\begin{aligned} \theta(a \square b) &= \theta(a) \bigcirc \theta(b) \\ \theta(a \boxtimes b) &= \theta(a) \otimes \theta(b) \end{aligned} \quad \text{for all } a, b \in R. \tag{6}$$

If we assume that errors are additive, such that $\square$ models the effects of system faults, then our results from groups carry over completely. The only difference is that ideals rather than normal subgroups are used to define quotient rings. An SSC thus defined is capable of protecting both ring operations.

A field $F = [\mathcal{F}; \square, \boxtimes, 0_\square]$ is a ring with the added property that nonzero elements form a group under $\boxtimes$. The results for rings still apply and we can determine systematic-separate codes by finding ideals. However, a field contains only trivial ideals, and the only SSC which may be defined is isomorphic to duplicating the original computation in the parity channel (equivalent to double modular redundancy). Lower complexity codes are impossible.

We have also been able to extend our technique to vector spaces, modules, and certain monoids. The basic results are the same and we are again guaranteed to find all systematic-separate coding schemes.

4. Examples

We now give two examples of our technique, one for a group, and another for a ring. Let G be the group of integers under addition. Normal subgroups of G are of the form $N = \{0, \pm M, \pm 2M, \ldots\}$ where M is an integer. The quotient group $\tilde{T} = G/N$ contains M unique cosets which we denote $\tilde{t}_0, \tilde{t}_1, \ldots, \tilde{t}_{M-1}$. $\tilde{t}_k$ is of the form $\{k, k \pm M, k \pm 2M, \ldots\}$. Adding two elements in G/N we find that $\tilde{t}_i \tilde{\bigcirc} \tilde{t}_j = \tilde{t}_{\langle i+j \rangle_M}$ where $\langle x \rangle_M$ denotes the remainder of x modulo M. Examining $\tilde{T}$, we see that it is isomorphic to the group T of integers $\{0, 1, \ldots, M-1\}$ under modulo M addition. This isomorphism is accomplished by mapping the coset $\tilde{t}_i$ to the integer i. The mapping from G to T is given by $t_g = \theta(g) = \langle g \rangle_M$, and the parity operation $t_i \bigcirc t_j = \langle t_i + t_j \rangle_M$. Thus, we have shown that the

only SSC capable of protecting integer addition is isomorphic to a residue checksum modulo an integer M.

If we assume that the parity channel is robust, then it is easy to determine which errors may be detected. For a given N, the kernel $K_\theta = N$. Thus, any error in the set $\{0, \pm M, \pm 2M, \ldots\}$ is undetectable. To be correctable, each error must have a unique syndrome. Since T is a finite group containing $M-1$ nonzero elements, we can never correct more than $M-1$ different errors.

We now know the form of the SSC and which types of errors may be detected and corrected. To finish the design, we must choose a specific modulus M which protects against the set of errors produced by hardware failures. If we expect errors to randomly corrupt digits of the result, then M must be chosen co-prime to the base of the number system used.

The second example which we give is the linear convolution $c[n] = a[n] * b[n]$ where $a[n]$ and $b[n]$ are both P-point sequences which are nonzero only in the interval $0 \leq n \leq P-1$. Assume that the samples are members of a field F, and denote by $F[x]$ the set of all polynomials in an indeterminate x with coefficients in F. The convolution can be performed in the polynomial ring $R = [F[x]; +, \times, 0]$ where $+$ and $\times$ denote ordinary polynomial addition and multiplication. Let $a(x) = \sum_{n=0}^{P-1} a[n]x^n$ and $b(x) = \sum_{n=0}^{P-1} b[n]x^n$. Then $c(x) = a(x) \times b(x)$ where $c(x) = \sum_{n=0}^{2P-2} c[n]x^n$. Thus, if we can protect this ring multiplication, we can protect the convolution.

We can protect polynomial multiplication with an SSC. The ideals of R are sets of polynomials of the form

$$N = \{g(x)M(x) \mid g(x) \in R\} \tag{7}$$

where $M(x) \in R$. We omit the derivation and proceed to the final result. For a given $M(x)$, the ring of parity symbols is isomorphic to $F[x]/M(x)$, the ring of polynomials modulo $M(x)$. Addition and multiplication in $F[x]/M(x)$ are ordinary polynomial addition and multiplication modulo $M(x)$. The parity symbols $t_a(x)$ and $t_b(x)$ are computed as follows:

$$t_a(x) = \langle a(x) \rangle_{M(x)} \qquad \text{and} \qquad t_b(x) = \langle b(x) \rangle_{M(x)} \tag{8}$$

where $\langle a(x) \rangle_{M(x)}$ denotes the remainder when $a(x)$ is divided by $M(x)$. The parity computation is

$$t_c(x) = \langle a(x) \times b(x) \rangle_{M(x)}. \tag{9}$$

Since the ideals of R are all of the form shown in (7), all systematic-separate codes for convolution are isomorphic to the one presented here. It is also possible to protect this operation with multiple parity channels, each utilizing a co-prime modulus $M_k(x)$. This is included in our framework since it is isomorphic to a single channel modulo $M(x) = \prod_k M_k(x)$.

To finish the design of the fault-tolerant convolution system, we must choose the polynomial $M(x)$ to protect against the expected set of errors. We can apply existing error coding techniques by noting that (8) is the standard method of encoding a systematic cyclic linear error-correcting code. Furthermore, fast algorithms for detecting and correcting errors exist [11].

5. Conclusion

In this paper, we presented a new and mathematically rigorous method of developing fault-tolerant systems which is based on group theory. We modeled computation as operations in an algebraic group, and were able to use theoretical results in group theory to identify possible codes. We feel that this work is significant because it is a precise method of modeling computation and adding redundancy. The main contribution of our technique is revealing the form which redundant information must take in systematic-separate codes. In many instances, we are able to identify all codes of this type, or even disprove the existence of any such codes. We hope that our approach will stimulate further research into more efficient fault-tolerant systems.

References

[1] T. R. N. Rao and E. Fujiwara, *Error-Control Coding for Computer Systems.* Englewood Cliffs, N.J.: Prentice-Hall, 1989.

[2] B. R. Musicus and W. S. Song, "Fault-tolerant digital signal processing via generalized likelihood ratio tests." Submitted to *IEEE Transactions on Signal Processing*, April 1990.

[3] K.-H. Huang and J. A. Abraham, "Algorithm-based fault tolerance for matrix operations," *IEEE Transactions on Computers*, vol. C-33, pp. 518–528, June 1984.

[4] P. E. Beckmann and B. R. Musicus, "Fault-tolerant round-robin A/D converter systems," *IEEE Transactions on Circuits and Systems*, vol. 38, December 1991. To appear.

[5] P. E. Beckmann and B. R. Musicus, "Fast fault-tolerant digital convolution via a Winograd algorithm." Submitted *IEEE Transactions on Signal Processing*, March 1991.

[6] G. R. Redinbo, "System level reliability in convolution computations," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1241–1252, August 1989.

[7] W. W. Peterson, "On checking an adder," *IBM Journal*, pp. 166–168, April 1958.

[8] H. L. Garner, "Error codes for arithmetic operations," *IRE Transactions on Electronic Computers*, vol. EC-15, pp. 763–770, October 1966.

[9] P. E. Beckmann and B. R. Musicus, "Fault-tolerant computation through algebraic homomorphisms." Manuscript in Progress. To be submitted for publication, December 1991.

[10] I. N. Herstein, *Topics in Algebra.* New York, NY: John Wiley and Sons, 1975.

[11] R. E. Blahut, *Theory and Practice of Error Control Codes.* Reading, MA.: Addison-Wesley Publishing Company, 1984.