THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

THE ALGEBRA OF A COMPUTER INTEGER ARITHMETIC SYSTEM

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

BY

LAWRENCE ANDREW OLDROYD

Norman, Oklahoma

1977

THE ALGEBRA OF A COMPUTER INTEGER ARITHMETIC SYSTEM

APPROVED BY

R. V. Andree

Irene Levy

Arthur Bernhart

John W. [illegible]

Harold Hinds

DISSERTATION COMMITTEE

ACKNOWLEDGEMENTS

I wish to recognize and thank a number of people for their help in the successful creation of this dissertation. First, by far, is Dr. Richard V. Andree, my advisor, without whose untiring assistance, advice and guidance this paper might never have come to be. Thanks also to my committee members, Dr. Arthur Bernhart, Dr. John Green, Dr. Harold Huneke and Dr. Gene Levy. A special note of gratitude goes to my typist, Claudia Embry, who did an excellent job with very difficult material.

TABLE OF CONTENTS

THE ALGEBRA OF A COMPUTER INTEGER ARITHMETIC SYSTEM

CHAPTER 1

INTRODUCTION

Most applied mathematical models assume a number system which is a field, or at least an integral domain. Models in science and engineering are generally based on the field of real numbers. Business applications often do not require such a sophisticated system, but do in general use the field of rational numbers or an integral domain. Calculations involving only whole numbers or integers may be performed in the ring of integers (an integral domain). Computer arithmetic systems were devised to simulate such arithmetics.

Computers do not use these theoretical mathematical systems. Physical and economic limitations force many constraints on computer arithmetic. Principal among these is a restriction to a fixed digital precision in positional notation; numbers must be represented by some fixed number of digits in a given radix. High-speed operation implies that computer arithmetic should be no more complicated than necessary. To simulate arithmetic of the real or rational field and of the ring of integers within these constraints, computers use two basic arithmetic forms: floating point and so-called "integer" systems.

Computer floating point arithmetic is used to simulate operations

in the real or rational fields. Numbers are represented in a form closely related to that of "scientific notation," using two components: a fixed point mantissa and an integer scaling operator, or exponent. Arithmetic in such a system is fairly complicated. Addition and subtraction require comparison of scaling operators, and repositioning of the mantissas relative to each other (aligning the "decimal point") before the operation can be performed. Rescaling may be needed afterwards. Multiplication and division are easier, but they also require operations on both components of the floating point numbers.

Computer "integer" systems derive their name from simulating operations of the ring of integers. Integer systems seem simpler than floating point systems. No scaling operator is used, no alignment of operands is required before the operation and no rescaling afterwards. This simplicity makes it the fastest arithmetic to perform. In fact, computer integer arithmetic is used to implement floating point operations, particularly on computers lacking built-in floating point hardware.† Furthermore, integer systems have the advantage of being exact. As long as a condition called overflow does not occur, computer integer arithmetic parallels arithmetic in the ring of integers. This is not true with floating point. These properties make integer systems most beneficial in environments, such as accounting systems, where only integer values are needed.

Integer arithmetic systems are achieving a position of greater importance lately. In small computers, particularly mini-computers and

†Even on computers with built-in floating point operations, these operations are performed using integer or integer-like arithmetic.

the newly emerging microprocessors, integer arithmetic is the only form available. Floating point arithmetic, a "must" for scientific computing, is performed by programmed routines using integer arithmetic. Despite this importance, and despite the fact that most computers offer integer arithmetic, study of these systems as mathematical algebraic systems has been limited. It is this important topic, the various systems of computer integer arithmetic, that is the subject of this dissertation.

The study will consider a major system of computer integer arithmetic called the radix complement system. Its basic structure as a modular ring of integers will be demonstrated. A division operation, as performed by computers, will then be investigated; this operation will be compared to other division operations defined on modular rings. The extension of algebraic order properties to this system will be considered; although a modular ring cannot have an order, certain properties of an algebra of inequalities can be shown. Consideration of the interaction of computer division with the usual ring operations will then lead to the development of a non-standard form for representing the rational numbers. This form may preview a new system for simulating rational arithmetic having advantages over the floating point system in some applications.

The first step in this investigation is one of definition. The radix complement system will be defined, along with its basic ring operations addition, subtraction and multiplication.

CHAPTER 2

THE RADIX COMPLEMENT SYSTEM

Computer arithmetic systems must be both fast and simple. Economic constraints give rise to three basic requirements for most computer arithmetic systems.

First, all elements of the system should be represented in the same format; this usually means they should all have some fixed number of digits m in a given radix representation, radix r.

Second, all digits in the representation should, as far as possible, be treated the same in the logic circuitry of the arithmetic unit. The primary significance here is that there should be no distinct sign "digit" appended to the elements, even though they may represent positive or negative integer quantities.

Finally, all arithmetic should be performed using as few and as simple operations as possible.

The number of digits in element representations is usually taken to be the number of digits that can be stored in one unit of computer memory, a "word." This may vary from 4 to 8 digits with mini- and micro-computers to more than 30 digits for large business and scientific computers. Radices used are usually some power of 2, because the logic circuitry of their implementation is binary. The most

common are binary (base 2), octal (base 8), and hexadecimal (base 16). The decimal radix (base 10) is not in general use on present computers.

A distinct sign "digit" for numbers has two disadvantages. First, it may require that a decision be made during arithmetic operations. Addition of operands having opposite signs may involve a different process than adding operands of the same sign, or it might be necessary to separately determine the sign of the result. Such a decision introduces unavoidable delay into the arithmetic process, violating the demand for speed. A distinguished sign digit may also require its own special logic circuitry, meaning greater logical complexity and expense for the arithmetic unit.

The desire to have a simple arithmetic unit creates the third demand. Hand calculation uses only the operations of addition and subtraction. Multiplication and division are performed using algorithms that repetitively apply one or the other of those operations. If a suitably simple method of encoding the additive inverse of numbers can be found, it is possible to do all arithmetic operations using only addition and negation (complementation); subtraction is done by adding the additive inverse, multiplication and division algorithmically.

The first two of these criteria are clearly satisfied by the system of integers, modulo b, where b is a positive integer. Z_b (or Z/bZ in another notation) may be thought to consist of the numbers 0, 1, 2, ... , b-1, with arithmetic performed modulo b. Lacking a positive class, this system has no need to append signs to any elements, and all elements can have the same number of digits, radix r, by including leading zeros as required. The most common computer inte-

ger systems in use today are modular, and have a simple technique for determining the additive inverse of an element.

The three principal computer integer arithmetic systems are the "sign and magnitude," the "radix complement," and the "diminished radix complement" systems. The first of these, in form, is rather like the number representations used in hand calculation. It uses a distinguished "digit" to represent the sign of its elements, with all the implied disadvantages, and will not be considered further. The radix and diminished radix complement system are the most common systems in use, particularly the radix complement system. On binary computers, they are known as the "2's complement" and "1's complement," respectively. Both are modular, but differ in the way in which the additive inverse of an element is determined, and in the logic circuitry required to perform addition. The diminished radix complement system uses a very simple method for obtaining the additive inverse of an element. However, addition of two elements is a two-step process and requires a decision before the final step may be completed. As before, a decision imbedded in an operation makes it more complex and slow. For this reason, the diminished radix complement system is much less popular than the radix complement system. The latter will be considered further.

Let C_m^r stand for the radix complement system, where m is the number of digits, and r is the radix. An element of the system will be represented by an m-dimensional vector $(d_{m-1}, d_{m-2}, \ldots, d_1, d_0)$ or simply as a string of digits $d_{m-1}d_{m-2}\ldots d_1d_0$, where each digit d_i is one of the numbers $0, 1, 2, \ldots, r-1$. As such, these elements

may be taken to be unsigned integers, with value given by $\sum_{i=0}^{m-1} d_i r^i$. This is how they will be treated for addition, subtraction, and multiplication; their appearance for division and to someone using the system will be somewhat different. Addition on this system is defined as unsigned radix addition, modulo r^m. That is, if A and B are two elements of C_m^r with unsigned values $\sum_{i=0}^{m-1} a_i r^i$ and $\sum_{i=0}^{m-1} b_i r^i$ respectively, their sum A + B is that element of C_m^r having unsigned value $\sum_{i=0}^{m-1} (a_i + b_i) r^i \bmod r^m$. This is the usual radix r integer addition with the result truncated to the least significant m digits. Indeed, in computers, only the right-most m digits of the sum may be formed. An important point in this system is that there is no distinguished digital position, that is no "sign" digit for any number; all digital positions are treated the same during addition. Furthermore, no allowance need be made for the addition of oppositely signed arguments, because there are no signs.

The element with all zero digits is an additive identity, and will be represented in C_m^r using the symbol 0. For each element A of this system, with unsigned value $\sum_{i=0}^{m-1} a_i r^i$, there corresponds an additive inverse, namely that element with unsigned value $r^m - \sum_{i=0}^{m-1} a_i r^i$. This is the radix complement of the element A, and is the "negative" of A. To avoid having a separate operation of subtraction, we take the difference $A \ominus B$ to be the element $A \oplus \overline{B}$, where $\overline{B}$ is the radix complement of B. The utility of this system depends on it being

faster and easier to use $\overline{B}$ than to have a separate subtraction operation.

The formation of the radix complement is relatively easy. It is found by complementing each digit d_i of an element to r-1 and adding 1 modulo r^m to the result; each d_i is replaced by the digit $r-1-d_i$ and 1 is added into the d_0 digital position, with carry propagation permitted and any carry beyond the m least significant digits ignored. To see that this process yields the radix complement, let A be an element of C_m^r with unsigned value $\sum_{i=0}^{m-1} a_i r^i$ and observe that

$$\sum_{i=0}^{m-1} (r-1-a_i)r^i + 1 = \sum_{i=0}^{m-1} (r-1)r^i + \sum_{i=0}^{m-1} a_i r^i = r^m - \sum_{i=0}^{m-1} a_i r^i.$$

In the case of radix 2, as used in most computers, this is simple; each digit is 0 or 1, and the complement is found by changing all zeros to ones and all ones to zeros, adding 1 to the least significant digital position. The carry propagation necessary in the formation of the radix complement does not usually slow arithmetic operations, because it normally occurs concurrently with the carry propagation in an addition. If the addition of 1 into the least significant digital position is not performed during complementation, the result is called the <u>diminished radix complement</u>.

Multiplication for the radix complement system is defined to be an integer multiplication, modulo r^m. Thus, for A and B in C_m^r, with unsigned values $\sum_{i=0}^{m-1} a_i r^i$ and $\sum_{i=0}^{m-1} b_i r^i$, their product A * B is that element having unsigned value $\left(\sum_{i=0}^{m-1} a_i r^i\right) \cdot \left(\sum_{i=0}^{m-1} b_i r^i\right) \bmod r^m$.

Again, the usual multiplication takes place by means of repeated addition and shifting of operands, after which the result is truncated to the m least significant digits. As with addition, it would be possible to form only the m right-most digits of the product, although this is not generally done on computers. The identity for multiplication is the element $00 \ldots 01$, and will be denoted in C_m^r by 1.

A couple of examples may now help clarify things.

Example 2.1

The elements of C_3^2 and C_2^3 are shown with their corresponding unsigned integer values. Note that the unsigned value of an element is just its value as an integer expressed in radix r.

C_3^2	Unsigned Value	C_2^3	Unsigned Value
000	0	00	0
001	1	01	1
010	2	02	2
011	3	10	3
100	4	11	4
101	5	12	5
110	6	20	6
111	7	21	7
		22	8

Example 2.2

The following table gives two elements from each of the systems C_4^2, C_4^{10}, and C_4^{16} along with their respective complements.

The System	The Element	Its Diminished Radix Complement		Its Radix Complement
C_4^2	0110	1001	+ 1 =	1010
C_4^2	1101	0010	+ 1 =	0011
C_4^{10}	2117	7882	+ 1 =	7883
C_4^{10}	9926	0073	+ 1 =	0074
C_4^{16}	02FC*	FD03	+ 1 =	FD04
C_4^{16}	D130*	2ECF	+ 1 =	2ED0

Isomorphism to $Z_r m$

The system described here is $Z_r m$, the integers modulo r^m, where numbers are expressed in radix r. To show this, it will help to formally define the unsigned value map from C_m^r to the integers Z.

Definition 2.3

Let η be the unsigned value map, $\eta: C_m^r \to Z$, defined by $\eta(A) = \sum_{i=0}^{m-1} a_i r^i$ for each element A in C_m^r, where $A = a_{m-1}a_{m-2}\cdots a_1 a_0$. Let $\eta^*: C_m^r \to Z_r m$ be defined by $\eta^*(A) = \eta(A) + r^m Z$ for each A in C_m^r. These maps are well-defined. Note that η^* is the composition of η with the natural map from Z to $Z_r m$. η is monic (one-to-one) because integer representations in a fixed positive radix r are unique. η^* is also monic; if $\eta^*(A) = \eta^*(B)$, then $\sum_{i=0}^{m-1} a_i r^i$ and $\sum_{i=0}^{m-1} b_i r^i$ differ

*The digits in the hexadecimal (radix 16) number system are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

by some multiple of r^m, and this can only happen if $a_i = b_i$ for each $i = 0, 1, \ldots, m-1$.

Furthermore, a counting argument will show that η^* must be epic (onto).

Three elementary theorems of algebra will be used. The notation used is as follows. To say that $(X,@)$ is a (semi) group means that X is a (semi) group under operation $@$. If $(X,@,\#)$ is a ring, then X is a ring, with $@$ the group operation on X and $\#$ the semi-group operation which distributes over $@$.

Theorem A.1

Suppose X is a set and $(Y,\cdot)$ is a semi-group. If $\eta: X \to Y$ is both monic and epic, and if an operation $*$ is defined on X by $a * b = \eta^{-1}(\eta(a) \cdot \eta(b))$ for all a and b in X, then $(X,*)$ is a semi-group and η is an isomorphism of X to Y.

Theorem A.2

Suppose X is a set and $(Y,+)$ is a group. If $\eta: X \to Y$ is both monic and epic, and if an operation $\oplus$ is defined on X by $a + b = \eta^{-1}(\eta(a) + \eta(b))$ for all a and b in X, then $(X,\oplus)$ is a group and η is a group isomorphism of X to Y.

These two combine to give the third theorem.

Theorem A.3

Suppose X is a set and $(Y,+,\cdot)$ is a ring. If $\eta: X \to Y$ is a group isomorphism of $(X,\oplus)$ to $(Y,+)$ as in Theorem A.2, and a semi-group isomorphism of $(X,*)$ to $(Y,\cdot)$ as in Theorem A.1, then $(X,\oplus,*)$ is a ring isomorphic to $(Y,+,\cdot)$ by η.

Consider now the set C_m^r, the modular ring of integers $Z_r m$, and the map $\eta^*: C_m^r \to Z_r m$. It has been noted that η^* is both monic and epic. The operations of addition and multiplication on C_m^r were defined with $A \oplus B = \eta^{-1}(\eta(A) + \eta(B) \bmod r^m)$ and $A * B = \eta^{-1}(\eta(A) + \eta(B) \bmod r^m)$, so that $A \oplus B = \eta^{*-1}(\eta^*(A) + \eta^*(B))$ and $A * B = \eta^{*-1}(\eta^*(A) \cdot \eta^*(B))$ for all A and B in C_m^r. Theorems A.1, A.2, and A.3 may thus be applied to give

Theorem 2.4

C_m^r is isomorphic to $Z_r m$ as a ring. The map η^* is the isomorphism.

For radix 2, the 2's complement number system, we have

Corollary 2.5

$C_m^2 \cong Z_2 m$ as rings.

The Signed Value Map

The radix complement number system was not developed so that computer integer arithmetic would be modular. The modular system was used because it allowed the fastest and most efficient arithmetic logic circuitry. The purpose of the computer integer system is to simulate, at least in part, the integer arithmetic system; the modular system, in its unsigned value form, does this rather poorly. Every non-zero element appears, under the map η, as a positive integer, so a number and its additive inverse both appear as positive integers, usually of different magnitudes. In the integers, a non-zero number and its additive inverse have the same magnitude and opposite signs. Thus, a signed value map is needed, with the property that for most elements of C_m^r, a

number and its negative are assigned integer values of the same magnitude and opposite signs. An observation about the unsigned values will indicate how this may be accomplished.

Lemma 2.6

Let A be an element of C_m^r, with additive inverse $\overline{A}$, and let $h = \frac{r^m}{2}$. If $\eta(A) < h$, then $\eta(\overline{A}) > h$. If $\eta(A) = h$, then $\eta(\overline{A}) = h$, and in this case $A = \overline{A}$.

Proof: These follow from the definition of the additive inverse, because η yields non-negative values, and $\eta(\overline{A}) = r^m - \eta(A)$. If $\eta(A) = \eta(\overline{A})$, the one-to-one property of η says that $A = \overline{A}$.

Lemma 2.6 indicates that the non-zero elements of C_m^r may be separated into two sets. The elements with unsigned values greater than $\frac{r^m}{2}$ are all additive inverses of the others, so this set with large unsigned values will be given negative signed values. Elements with unsigned value less than $\frac{r^m}{2}$ will be positive, and take their unsigned value for their signed value. Each element A with unsigned value greater than $\frac{r^m}{2}$ will take $-\eta(\overline{A})$ as its signed value. It will be shown that for each element A in C_m^r, the signed values of A and $\overline{A}$ defined in this way have the same magnitude and opposite signs.

The foregoing is not valid for an element of C_m^r with unsigned value $\frac{r^m}{2}$. Can such an element exist? The answer depends on the radix r. If r is odd, then r^m is odd, and $\frac{r^m}{2}$ is not an integer value; there can be no element having unsigned value $\frac{r^m}{2}$. If r is even (the case for all major computers now in use), then r^m is even, and there is exactly one element of C_m^r with this unsigned value. To

determine whether it should have positive or negative signed value, consider the radix r representation of this element: $(\frac{r}{2}, 0, \ldots, 0)$. Note that $\frac{r}{2}$ is an integer because r is even. All other elements with leading digit $\frac{r}{2}$ have unsigned values greater than $\frac{r^m}{2}$ and negative signed values. It is prudent then to give this element the signed value $-\frac{r^m}{2}$, recognizing that it is its own additive inverse, and there will be no element having the complementary positive signed value.

It is now possible to formally define the signed value map.

Definition 2.7

The signed value map $\phi: C_m^r \to Z$ is given by

$$\phi(A) = \begin{cases} \eta(A) & \text{if } \eta(A) < \frac{r^m}{2}, \\ \eta(A) - r^m & \text{if } \eta(A) \geq \frac{r^m}{2}, \end{cases}$$

for each element A in C_m^r.

Note that $\eta(A) - r^m = -\eta(\overline{A})$. The next theorem shows that $\phi(A)$ and $\phi(\overline{A})$ have the same magnitude and opposite signs:

Theorem 2.8

Let A be an element of C_m^r and let $\overline{A}$ be its radix complement. If $\eta(A) \neq \frac{r^m}{2}$, then $\phi(A) = -\phi(\overline{A})$. If $\eta(A) = \frac{r^m}{2}$, then $\phi(A) = \phi(\overline{A})$.

Proof: Using the definition of ϕ,

$$\phi(\overline{A}) = \begin{cases} \eta(\overline{A}) & \text{if } \eta(\overline{A}) < \frac{r^m}{2}, \\ \eta(\overline{A}) - r^m & \text{if } \eta(\overline{A}) \geq \frac{r^m}{2}. \end{cases}$$

From previous remarks, $\eta(\bar{A}) < \frac{r^m}{2}$ means $\eta(A) > \frac{r^m}{2}$, and $\eta(\bar{A}) \geq \frac{r^m}{2}$ means $\eta(A) \leq \frac{r^m}{2}$. Since $\eta(\bar{A}) = r^m - \eta(A)$, the equation may be written

$$\phi(\bar{A}) = \begin{cases} -\eta(A) & \text{if} \quad \eta(A) \leq \frac{r^m}{2}, \\ r^m - \eta(A) & \text{if} \quad \eta(A) > \frac{r^m}{2}. \end{cases}$$

Comparing this with Definition 2.7 yields the conclusion.

The terms "positive" and "negative" will be applied to elements of C_m^r according to their signed values. It should, however, be carefully noted that C_m^r is not divided into positive and negative classes as is the system of integers. The radix complement system is modular and cannot have a positive class. This implies that there is no order relation on C_m^r as there is on Z. Elements of C_m^r may, nevertheless, be ranked (not ordered) by their signed integer values as "less than," "greater than," and so forth. This will be indicated using the FORTRAN relational operator symbols .GT., .GE., .EQ., .LE., .LT., and .NE., which correspond to the integer relations $>$, $\geq$, $=$, $\leq$, $<$, and $\neq$, respectively. It will also be useful to speak of the "absolute value" of elements of C_m^r. This will be taken to mean the absolute value of the signed value given by the map ϕ.

Example 2.9

Consider the systems C_3^2 and C_2^3 of Example 2.1, and their signed values.

C_3^2	signed value	C_2^3	signed value
		11	4
011	3	10	3
010	2	02	2
001	1	01	1
000	0	00	0
111	-1	22	-1
110	-2	21	-2
101	-3	20	-3
100	-4	12	-4

The radix complement system for an odd radix r will have as many positive as negative elements, taking zero as neither positive nor negative. The number of non-zero elements, $r^m - 1$, is even because r is odd. The signed value map will give exactly half of these positive values and the rest negative values. An arithmetic system having the same numbers of positive and negative elements will be called balanced. For radix r even, the radix complement system is not balanced; it has an odd number of non-zero elements. This system would be balanced, except for the one element which is its own additive inverse. Because that element was taken to have negative signed value, there is one more negative element than positive elements. Example 2.8 illustrates both the unbalanced and balanced radix complement systems. The diminished radix complement system is balanced for even radix r, and unbalanced for odd radix. The sign and magnitude system is balanced for any radix.

It is often important to test the sign of a number without considering its magnitude. As usual, this test should be as simple as

possible. In the case of an even radix r, the definition of the signed value map ϕ ensures that any element having a digit $\frac{r}{2}$ or larger as its most significant digit will be negative. Thus, the value of a single digit may be used to evaluate the sign of an element. A system having this property is said to permit a single digit sign test. If the radix is odd, the element $(\frac{r-1}{2}, \ldots, \frac{r-1}{2}, \frac{r-1}{2})$ is positive, but the element $(\frac{r-1}{2}, \ldots, \frac{r-1}{2}, \frac{r+1}{2})$ is negative. This follows because

$$\sum_{i=0}^{m-1} (\tfrac{r-1}{2}) r^i = \frac{r^m - 1}{2} < \frac{r^m}{2} \qquad \text{and} \qquad \sum_{i=0}^{m-1} (\tfrac{r-1}{2}) r^i + (\tfrac{r+1}{2}) = \frac{r^m + 1}{2} > \frac{r^m}{2},$$

which may be established using induction on m. Thus, the radix complement system for odd radix does not permit a single digit sign test. The diminished radix complement system has the same property; it permits a single digit sign test only for even radices. Since the sign and magnitude system has a distinguished sign "digit," it will always permit this test.

The radix complement system cannot be balanced and permit a single digit sign test, simultaneously. Of these two, the sign test is far more significant concerning the logic circuit complexity of a computer, suggesting that even radix systems would be more popular. Indeed, the most significant radices are all even, but this is not the reason; even radices are used because binary (on-off) logic circuitry is the simplest available. The sign test in a radix 2 system is particularly simple; the most significant digit is either 0 or 1, with 0 indicating positive and 1 indicating negative. If the radix is a power of 2, such as radix 8 or radix 16, the sign test may be equally simple. In such a system, the digits are usually coded as binary numbers, and the sign test is per-

formed as in radix 2 by inspecting the most significant bit (binary digit) of the most significant digit.

Using the single digit sign test in the even radix case, it is possible to give a simpler expression for the signed value function:

$$\phi(A) = \begin{cases} \sum_{i=0}^{m-1} a_i r^i & \text{if } a_{m-1} < \frac{r}{2}, \\ \sum_{i=0}^{m-1} a_i r^i - r^m & \text{if } a_{m-1} \geq \frac{r}{2} \end{cases}$$

where $A = (a_{m-1}, \ldots, a_2, a_1)$ and r is even. With a binary system, this becomes

$$\phi(A) = -a_{m-1} 2^{m-1} + \sum_{i=0}^{m-2} a_i 2^i \quad .$$

The elements having largest positive and negative magnitudes will be used quite often. For this reason, they will be denoted by P_m^r and N_m^r, respectively. If the radix is unimportant, or understood from context, these will be simplified to Pm and Nm. The definition of positive and negative for this modular system implies that $Pm \oplus 1 = Nm$. For even radix,

$$Pm = (\tfrac{r}{2} - 1, r-1, \ldots, r-1) \quad \text{with} \quad \phi(Pm) = \frac{r^m}{2} - 1,$$

and

$$Nm = (\tfrac{r}{2}, 0, \ldots, 0) \quad \text{with} \quad \phi(Nm) = -\frac{r^m}{2} .$$

For odd radix,

$$Pm = (\tfrac{r-1}{2}, \ldots, \tfrac{r-1}{2}, \tfrac{r-1}{2}) \quad \text{with} \quad \phi(Pm) = \tfrac{r^m-1}{2},$$

and

$$Nm = (\tfrac{r-1}{2}, \ldots, \tfrac{r-1}{2}, \tfrac{r+1}{2}) \quad \text{with} \quad \phi(Nm) = -\tfrac{r^m-1}{2}.$$

For radix 2, these become $Pm = (0, 1, \ldots, 1)$ with $\phi(Pm) = 2^{m-1} - 1$, and $Nm = (1, 0, \ldots, 0)$ with $\phi(Nm) = -2^{m-1}$.

It is important to note that the set of signed values produced by ϕ is equivalent, modulo r^m, to the set of unsigned values produced by η. For each element A in C_m^r, $\phi(A) \equiv \eta(A) \bmod r^m$. This is clear from Definition 2.7. Let $S_m^r = \{\phi(A)\colon A \text{ is an element of } C_m^r\}$ be the set of signed values. Because $S_m^r \subset Z$, the usual integer operations $+$, $-$, and $\cdot$ apply to elements of S_m^r, although they are not in general closed on the set. The arithmetic operations modulo r^m may be defined as follows.

Definition 2.10

Let A and B be elements of S_m^r. Define operations $\oplus$, $\ominus$ and $*$ by

$$A \oplus B = A + B + ir^m,$$
$$A \ominus B = A - B + jr^m,$$
and
$$A * B = A \cdot B + kr^m,$$

where integers i, j, and k are chosen so that

$$\phi(Nm) \leq A \oplus B \leq \phi(Pm),$$
$$\phi(Nm) \leq A \ominus B \leq \phi(Pm),$$
and
$$\phi(Nm) \leq A * B \leq \phi(Pm), \qquad \text{respectively.}$$

The set S_m^r, together with the operations $\oplus$, $\ominus$, and $*$, are one representation of $Z_r m$, so for these operations, $S_m^r \cong Z_r m$. Because $Z_r m \cong C_m^r$, it follows that $S_m^r \cong C_m^r$. This gives the next theorem.

Theorem 2.11

Let S_m^r be the set of signed integer values of elements of C_m^r, and let $\oplus$ and $*$ be the modular operations on S_m^r given in Definition 2.10. Then the ring $(S_m^r, \oplus, *)$ is isomorphic to the ring $(C_m^r, \oplus, *)$, and the signed value map ϕ is the isomorphism.

Elements of C_m^r are usually seen as signed integers, particularly by users of a high-level programming language. The isomorphism means that it will not generally be necessary to distinguish between an element of C_m^r and the corresponding element of S_m^r; an expression over C_m^r may be considered an expression over S_m^r. If such distinction is needed, the map ϕ may be used.

There are thus two sets of operations defined for elements of S_m^r: the usual integer operations $+$, $-$, and $\cdot$, and the corresponding modular operation $\oplus$, $\ominus$, and $*$. The integer operations will not be closed on S_m^r. An expression over S_m^r may involve operations from either set, but for simplicity, not from both sets.

Overflow

The map $\phi: C_m^r \to Z$ is one-to-one, but not a ring morphism for the integer operations; it will fail in many cases to preserve integer addition or multiplication. There are, however, many instances in which ϕ appears to act like one. In particular, if A and B are elements of C_m^r and $\phi(Nm) \leq \phi(A) + \phi(B) \leq \phi(Pm)$, then

$\phi(A \oplus B) = \phi(A) + \phi(B)$. Similarly, if $\phi(Nm) \leq \phi(A) \cdot \phi(B) \leq \phi(Pm)$, then $\phi(A*B) = \phi(A) \cdot \phi(B)$. These both follow because ϕ is one-to-one. At such times, C_m^r, through its signed values, exactly simulates the usual signed integer system. This is of such importance to the normal user of the computer integer system that those situations where ϕ fails as a morphism are identified by a special term overflow.

Definition 2.12

Let A and B be elements of C_m^r, and let $@_c$ stand for any one of the operations $\oplus$, $\ominus$, or * on C_m^r, with $@_z$ the corresponding operation +, -, or $\cdot$ on Z. If $\phi(A @_c B) \neq \phi(A) @_z \phi(B)$, then the operation $A @_c B$ is said to overflow.

Theorem 2.11

The operation $A @_c B$ on C_m^r overflows if and only if the corresponding operation $\phi(A) @_z \phi(B)$ on Z gives a value greater than $\phi(Pm)$ or less than $\phi(Nm)$.

Proof: This follows from the remarks above, because ϕ is monic. If $\phi(A) @_z \phi(B) > \phi(Pm)$, it is called positive overflow; the other case is called negative overflow.

Computer Integer Arithmetic

Most binary computers use 2's complement arithmetic. The system C_m^2 is isomorphic to Z_{2^m}. Does this mean that a binary computer using 2's complement arithmetic and having m bits per word functions as an arithmetic system isomorphic to Z_{2^m}? The answer is no, but can be taken for yes under one very simple constraint.

The reason for the "no" answer can be illustrated using the IBM System/360 integer arithmetic system. IBM/360 binary addition is modular; the problem arises with multiplication. In the integers, the product of the two numbers having at most m digits each will have at most $2m$ digits, and this upper bound on the number of digits can be attained. For example, in radix 10, the product of 9999 and 9999 is 99899001, an eight-digit result produced by multiplying two four-digit numbers. Binary integer multiplication on the IBM/360 actually works this way; multiplication of two m bit 2's complement numbers produce a $2m$ bit 2's complement result. (The number of digits m for the IBM/360 may be either 16 or 32.) This is not the multiplication operation defined for the C_m^2 system.

However, this view is looking too closely into the machine, and not into how the machine's arithmetic is used. For higher-level, user-oriented computer languages, such as FORTRAN, the machine's integer arithmetic results are truncated to the lowest order m bits before being used in subsequent operations. Thus, the integer arithmetic system used in these higher-level languages is indeed the radix complement system C_m^2. This is the restriction under which there is an arithmetic system isomorphic to Z_2m.

This completes the definition of the radix complement system with the operations addition, subtraction, and multiplication. The fourth elementary arithmetic operation, division, will be considered next.

CHAPTER 3

DIVISION

The division operation, defined on the ring of integers, Z, is a distinct third operation, only partly related to addition and multiplication. In a field, division is taken to be the "inverse" of multiplication in exactly the same sense as subtraction is the "inverse" of addition. However, except in trivial instances, multiplicative inverses do not exist in Z, so it does not have this inherent division operation. Lacking such a constraint, it is possible to define a number of different operations that present some characteristics of field division. One of the most basic of these is the standard "long division" of elementary arithmetic, modified to permit signed operands. This operation approximates division in the field of rational numbers by producing the integer part of the field quotient. A very important advantage is associated with this division; the algorithm used to perform it is easily implemented using the ring operation addition and is essentially the same as the algorithm used for floating point division. For that reason, it will be the division operation defined on C_m^r.

Division in Z

The development begins with the ring of integers. Part of it will be done using rational numbers, and confusion over division sym-

bols could result. To avoid this, the symbol $\div$ or the horizontal fraction bar — will indicate the rational quotient, or the rational number determined by the ratio of two integers. The slash mark / will be used for division on the integers.

Definition 3.1

For integers a and b, with $b \neq 0$, the primary quotient a/b is given by

$$a/b = \begin{cases} \left[\frac{|a|}{|b|}\right] & \text{if a and b have the same sign} \\ -\left[\frac{|a|}{|b|}\right] & \text{if a and b have different signs.} \end{cases}$$

The greatest integer function [x] for a rational number $x \geq 0$ is defined to be the greatest integer less than or equal to x. The primary quotient may be considered in the following way: express the rational number $\frac{a}{b}$ in signed radix r digital form, and truncate the fractional part. Alternately, if $\frac{a}{b}$ is expressed in mixed number form, where the fraction has magnitude less than one, the primary quotient represents the integer part.

The next theorem is basic for much that follows. It characterizes this quotient.

Theorem 3.2 (the primary division theorem)

Let a and b be integers, and q be their primary quotient. Then there exists an integer r, with $|r| < |b|$ and r either zero or having the same sign as a, such that $a = q \cdot b + r$.

Proof: $q = 0$ if and only if $|a| < |b|$. Thus, if $q = 0$, take $r = a$. If $q \neq 0$, then $q \cdot b$ will have the same sign as a, but

with $|a| - |b| < |q \cdot b| < |a|$. In this case, take $r = a - q \cdot b$.

The integer r is called the primary remainder, and is related to the fractional part when $\frac{a}{b}$ is expressed in mixed number form as $q + \frac{r}{b}$.

It will frequently be necessary to test the sign of a number, and for this it will help to have a special "sign" function. Define $s: Z \to \{-1, 0, 1\}$ by

$$s(a) = \begin{cases} 1 & \text{if } a > 0, \\ 0 & \text{if } a = 0, \\ & \text{and} \\ -1 & \text{if } a < 0. \end{cases}$$

Extending s to the radix complement system is done using the signed value map; $s(A) = s(\phi(A))$ for each element A in C_m^r.

The utility of the primary division algorithm in studying division arises because the representation it provides is unique.

Theorem 3.3

Let a and b be integers, with primary quotient $q = a/b$ and remainder r. If c and d are integers such that $a = c \cdot d + d$ with $|d| < |b|$ and either $d = 0$ or $s(d) = s(a)$, then $c = q$ and $d = r$.

Proof: The primary division algorithm gives $a = q \cdot b + r$, where $|r| < |b|$ and either $r = 0$ or $s(r) = s(a)$. Thus, $q \cdot b + r = c \cdot b + d$, which may be written as $(q - c) \cdot b = d - r$. If $a = 0$, then clearly $q = r = c = d = 0$. If either $d = 0$ or $r = 0$, then $|d - r| < |b|$, so $d = r = 0$ since the left side of the equation is divisible by b. This means that $q = c$. If neither d nor r is zero, then $s(d) = s(r)$,

so once again $|d - r| < |b|$, implying that $d = r$ and $q = c$.

Primary division resembles Euclidean division on Z. The integers are a Euclidean domain, and have a division defined as follows: if a and b are integers, $b \neq 0$, then there exist integers q and r, with $0 \leq r < |b|$, such that $a = q \cdot b + r$. The difference between this and primary division is that the Euclidean remainder is non-negative, while the other remainder takes the sign of the dividend. If the primary remainder is non-negative, the two quotients are the same; if it is negative, they differ by one.

Theorem 3.4

Let q and r be the primary quotient and remainder for integers a and b, and let q^e and r^e be the Euclidean quotient and remainder. Then

a) if $r \geq 0$, then $q = q^e$ and $r = r^e$,

b) if $r < 0$ and $b > 0$, then $q = q^e + 1$ and $r = r^e - b$,

and

c) if $r < 0$ and $b < 0$, then $q = q^e - 1$ and $r = r^e + b$.

Proof: a) This is clear from the definitions of the two algorithms.

b) With $r < 0$, the primary division form $a = q \cdot b + r$ is not quite correct for Euclidean division. However, if $b > 0$, then $0 < r + b < |b|$. Thus $a = (q - 1) \cdot b + (r + b)$ is the Euclidean form, with $q^e = q - 1$ and $r^e = r + b$.

c) This is similar to b), except $0 < r - b < |b|$. Then the Euclidean form is $a = (q + 1) \cdot b + (r - b)$, so $q^e = q + 1$ and $r^e = r - b$.

Division in C_m^r

Division in the radix complement system will be defined to be the restriction of the primary division for Z to the set S_m^r of signed integer representatives of C_m^r.

Definition 3.5

Let $\phi: C_m^r \to Z$ be the signed value map of Chapter 2, and let $S_m^r = \phi(C_m^r)$. Note that $S_m^r \subset Z$. If A and B are elements of C_m^r such that the primary quotient $\phi(A)/\phi(B)$ is in S_m^r, then the radix complement quotient A/B is given by

$$A/B = \phi^{-1}(\phi(A)/\phi(B)).$$

With this definition, the map ϕ is not only an isomorphism from $(C_m^r, \oplus, *)$ to $(S_m^r, \oplus, *)$, but a morphism for division as well, with $\phi(A/B) = \phi(A)/\phi(B)$. Elements of C_m^r are almost always used in their signed value form, and the quotient in C_m^r is the quotient of signed values. Because of this, the same symbol is used for division in C_m^r as for the primary quotient in Z. If distinction between the two operations is required, the type of operand will determine the system in question. For this to be effective, the following conventions will be used; elements of C_m^r or S_m^r will be represented by capital letters, and integers by lower case letters. Elements expressed as $\phi(X)$ represent integers, of course, and because this notation will become cumbersome at times, the underline symbol $\underline{X}$ may be used for $\phi(X)$.

The definition suggests that there may be elements A and B in C_m^r for which A/B is not defined. This is indeed the case, but there is only one exception other than the usual restriction to $B \neq 0$.

To see that this is true, suppose A and B are elements of C_m^r, with $B \neq 0$. Then $|\phi(A)/\phi(B)| \leq |\phi(A)|$, with equality only if $|B| = 1$. This means

$$-|\phi(A)| \leq \phi(A)/\phi(B) \leq |\phi(A)|,$$

and, in particular,

$$-|\phi(Nm)| \leq \phi(A)/\phi(B) \leq |\phi(Nm)|.$$

Since $-|\phi(Nm)| = \phi(Nm)$, this may be written

(1) $$\phi(Nm) \leq \phi(A)/\phi(B) \leq |\phi(Nm)|.$$

If C_m^r is balanced, $|\phi(Nm)| = |\phi(Pm)| = \phi(Pm)$, and (1) becomes

$$\phi(Nm) \leq \phi(A)/\phi(B) \leq \phi(Pm).$$

The set S_m^r is defined as $S_m^r = \{x \in Z: \phi(Nm) \leq x \leq \phi(Pm)\}$, so the inequality implies that $\phi(A)/\phi(B)$ is in S_m^r. Thus, A/B is defined for all A and B, with $B \neq 0$, in a balanced system.

If C_m^r is not balanced, then $|\phi(Nm)| = \phi(Pm) + 1$, so (1) becomes

$$\phi(Nm) \leq \phi(A)/\phi(B) \leq \phi(Pm) + 1.$$

If $\phi(A)/\phi(B) \leq \phi(Pm)$, then A/B is defined as shown above, but if $\phi(A)/\phi(B) = \phi(Pm) + 1$, then A/B is not defined. This can only happen with $A = Nm$ and $B = -1$. Therefore, A/B is defined for all A and B, $B \neq 0$, except when $A = Nm$ and $B = -1$. This completes the proof of

Theorem 3.6

If C_m^r is balanced (radix r odd), then A/B is defined for all A and B in C_m^r, with $B \neq 0$.

If C_m^r is not balanced (radix r even), then A/B is defined for all A and B in C_m^r, with $B \neq 0$, except when $A = Nm$ and $B = -1$.

In the 2's complement arithmetic on the IBM/360, an attempt to divide -2147483648 ($\phi(N_{32}^2) = -2^{31} = -2147483648$) by -1 produces a "divide exception;" the result would be 2147483648, a positive number too large to be a signed value ($\phi(P_{32}^2) = 2^{31} - 1 = 2147483647$) for any element of C_{32}^2. With the operations of addition, subtraction, and multiplication, this could not happen, because the result modulo 2^{32} would be produced; division is not a modular operation. The division algorithm finds the quotient by a process of repeated subtraction, and essentially keeps count of the number of subtractions performed; the count becomes the quotient. When the number of subtractions, taken with the correct algebraic sign, no longer represents a signed value for any element in C_{32}^2, the division algorithm fails.

This criterion for division algorithm failure is needed because the IBM/360 actually uses a 64 bit divisor. It is quite easy to have a quotient that cannot be expressed in 32 bits. In a higher-level programming language like FORTRAN, the 32-bit dividend argument (an element of C_{32}^2) is converted to a 64-bit dividend having the same signed value. Thus, only one exceptional case occurs for a non-zero divisor in C_{32}^2.

IBM/360 FORTRAN also allows the use of 16-bit integers, so that arithmetic may be performed in C_{16}^2. In this system, the only division exceptions recognized are attempts to divide by zero. Here, $N_{16}^2/(-1) = N_{16}^2$.

The reason that this occurs is that division on the IBM/360 cannot be performed with 16-bit operands; the dividend must have 64-bits and the divisor 32. To perform division in C_{16}^2, the arguments are first converted to 64 and 32-bit 2's complement numbers having the same signed values, respectively. The 32-bit result is truncated to the 16 least significant bits, reducing it modulo 2^{16}. Since $\phi(N_{16}^2) = -2^{15}$, dividing this number by -1 yields 2^{15} as the signed value of an element in C_{32}^2. This number reduced modulo 2^{16} gives -2^{15} as its signed value in C_{16}^2, making the quotient N_{16}^2.

The Division Theorem and Uniqueness in C_m^r

The development of the primary division theorem and uniqueness theorem for Z will not carry over to C_m^r. The division theorem will be the same for those cases where the C_m^r quotient is defined, but uniqueness is generally not obtained.

Theorem 3.7 (the division theorem in C_m^r)

Let A and B be elements of C_m^r, and let $Q = A/B$ be their quotient. Then there exists an element R in C_m^r, with $|R| < |B|$ and R either zero or having the same sign as A, such that $A = Q*B \oplus R$.

Proof: Note that $\phi(Q)\phi(B) = \phi(Q*B)$, so there is no overflow in this multiplication. The proof then follows exactly as for the integers. R is called the remainder.

The form $A = Q*B \oplus R$ will not be unique, but if $Q = A/B$, then the remainder R will be unique.

Theorem 3.8

Let A, B, C and Q be elements of C_m^r, such that $Q = A/B$ and

$A = Q*B \oplus C$. If R is the remainder from A/B, then $C = R$.

Proof: The division theorem in C_m^r gives $A = Q*B \oplus R$. Thus $Q*B \oplus C = Q*B \oplus R$ and $C = R$.

Distinct values of Q may satisfy $A = Q*B \oplus R$ with given A, B and R in C_m^r. Indeed, for fixed A and B, solutions having distinct values of Q and distinct values of R exist. This may be seen in the following example from the C_4^2 system.

Example 3.9

The elements in the C_4^2 system are $-8, -7, \ldots, -1, 0, 1, \ldots, 7$. Let $A = 7$ and $B = 4$. The division theorem for A/B gives $Q = 1$ and $R = 3$, with $7 = 1*4 \oplus 3$. However, it is also true that $7 = 5*4 \oplus 3$ and $7 = (-7)*4 \oplus 3$. For the integers, $7 = 1\cdot4 + 3$ is the unique representation in this form, but in C_4^2, there is no unique representation.

If $A = 7$ and $B = 5$ in C_4^2, then $Q = 1$ and $R = 2$, with $7 = 1*5 \oplus 2$. However, $7 = 4*5 \oplus 3$ also. Thus $A = Q*B \oplus R$ and $A = C*B \oplus D$, while $Q \neq C$ and $R \neq D$. Note that $D = 3$ satisfies the conditions for a remainder term, with the signs of 3 and 7 the same and $|3| < |5|$.

The failure to exhibit uniqueness comes from the modular structure of C_m^r. Its explanation requires some elementary number theory.

It is well-known that for a p-modular system, a linear congruence $xu \equiv v \bmod p$ may have zero, one or multiple solutions for x. If $g = (u,p)$ is the greatest common divisor of u and p, and $g|v$, then there are g distinct solutions to the congruence. If $g\nmid v$,

there are no solutions. It is also true that when $g|p$, there are exactly p/g elements in the p-modular system which are divisible by g.

Consider now elements A and B of C_m^r. Let $Q = A/B$ and R be the remainder, so $A = Q*B \oplus R$. Suppose X and Y are elements of C_m^r such that $A = X*B \oplus Y$. Then $(Q \ominus X)*B = Y \ominus R$. In the integers this becomes $(Q - X)B = Y - R \bmod r^m$. Let $g = (B, r^m)$. For each Y such that $g|(Y - R)$, there are exactly g distinct values of X such that $A = X*B \oplus Y$. If $g\nmid(Y - R)$, then there is no X such that $A = X*B \oplus Y$. Note that the value of $Y - R$ varies over all of C_m^r as Y varies over C_m^r. Since there are exactly r^m/g elements of C_m^r which are divisible by this g, there are r^m/g distinct values of Y for which $g|(Y - R)$. This means there are $g\cdot(r^m/g) = r^m$ pairs X, Y such that $A = X*B \oplus Y$. This proves the following theorem.

Theorem 3.10

Let A and B be elements of C_m^r. Then there are exactly r^m pairs X, Y of elements from C_m^r such that $A = X*B \oplus Y$.

The lack of the uniqueness property in C_m^r will not be a handicap. Expressions over C_m^r may be translated to S_m^r, and the modular operations to integer operations using Definition 2.10. The uniqueness property in Z will establish results for C_m^r.

As with multiplication, the division operation on a computer usually gives results which are not strictly in the C_m^r system. The computer produces both quotient and remainder when A/B is performed. Closure occurs in higher-level programming languages because only the quotient is kept. The remainder is then accessible by calculation.

The four basic arithmetic operations on the radix complement system are now defined. Addition, complementation (subtraction), and multiplication are modular ring operations and are well understood. Division is not modular and requires further consideration.

CHAPTER 4

A STUDY OF DIVISION ON C_m^r

Division on the radix complement system is a distinct third operation. It is not reversible by multiplication, as in a field. In general, $(A*B)/B \neq A$ and $(A/B)*B \neq A$. This operation will be considered from three viewpoints.

First, division is preserved by the signed value map $\phi: C_m^r \to Z$, which is an isomorphism from $(C_m^r, \oplus, *)$ to $(S_m^r, \oplus, *)$, with $S_m^r \subset Z$. Going the other way, the natural ring homomorphism $\xi: Z \to Z_r m$ may be composed with an isomorphism between $Z_r m$ and C_m^r to give a ring homomorphism $\psi: Z \to C_m^r$. Will this map also preserve division? The answer is <u>no</u>, and in fact, there is no extension of Z for which a ring homomorphism to C_m^r will preserve division. It is, however, possible to define a new division on Z, which will be preserved by ψ.

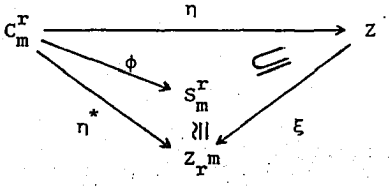
The second viewpoint is a comparison to a division operation defined on $Z_r m$. This operation may be defined using the linear congruence $BX \equiv A \bmod r^m$. When the congruence has a unique solution, this solution is taken as the quotient "A divided by B." Since $C_m^r \cong Z_r m$, how does this division relate to division on C_m^r? The relation is not close; modular division is not always defined, and even

when it is, the corresponding C_m^r quotient must be exact (remainder zero) before the two quotients agree. Furthermore, modular division may be "un-done" by multiplication, but this is possible for C_m^r division only if exact. Reciprocals exist for some elements of $Z_r m$, while only trivial reciprocals are non-zero in C_m^r.

An evaluation of classical group axioms, applied to the operation of division on C_m^r, constitutes the third approach. For this, the system $(C_m^r, /)$ will be studied. It is not a group, or even a semi-group; most of the group axioms fail for division on C_m^r.

Comparison to Division in Z

An overview of the entire setting may help. There is the unsigned value map $\eta: C_m^r \to Z$, which is one-to-one, with its associated map $\eta^*: C_m^r \to Z_r m$, a ring isomorphism. The signed value map $\phi: C_m^r \to Z$ is also one-to-one, and a ring isomorphism to its image S_m^r when operations $\oplus$ and $*$ are defined modulo r^m on S_m^r. There is also the natural ring homomorphism $\xi: Z \to Z_r m$. These are shown in the diagram below.



From the definition of η^*, both $\xi \circ \eta = \eta^*$ and $\xi \circ \phi = \eta^*$. The homomorphism ξ is the modulo r^m transformation, so η and ϕ are equivalent maps, modulo r^m.

The map η^* is a ring isomorphism, so the composition $\psi = \eta^{*-1} \circ \xi$ is a ring homomorphism. This means

$$\psi(a + b) = \psi(a) \oplus \psi(b) \quad \text{and} \quad \psi(ab) = \psi(a) * \psi(b)$$

for all integers a and b; the homomorphism preserves both addition (and subtraction) and multiplication. However, ψ does not preserve division, and is thus not a morphism for division. This is seen in the following example.

Example 4.1

Let Nm and Pm be the maximal negative and positive elements of C_m^r, respectively, and assume $|Nm| > 2$. Let a and b be integers, with $a = Pm + 1$ and $b = 2$. Pm represents its own signed value. Then $0 < a/b < Pm$, so $\psi(a / b) > 0$. But $\psi(a) = Nm$ and $\psi(b) = 2$, so $\psi(a) / \psi(b) < 0$, and $\psi(a) / \psi(b) \neq \psi(a / b)$. This shows ψ is not a morphism for division.

Would it then be possible to imbed Z in another system T with a morphism from T to C_m^r preserving all operations? This is not possible, because any extension of Z which preserved division and the ring operations would carry with it the same counter-examples used in Example 4.1. This is formalized in the next theorem and corollary.

Theorem 4.2

Suppose T is a system with maps $\mu: Z \to T$ and $\tau: T \to C_m^r$ satisfying these three properties:

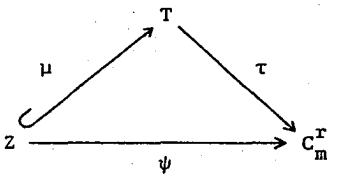
1) μ is an imbedding of Z into T that preserves division,

2) τ is a morphism from T to C_m^r that preserves division, and

3) $\tau \circ \mu = \psi$.

Then ψ is a morphism for division.

Note: Hypothesis 3) says that the results of operations in Z should be the same when viewed in C_m^r, regardless of whether they come directly from Z or indirectly through T. The following diagrams the relations between Z, T, and C_m^r.



Proof of theorem: Let a and b be elements of Z, $b \neq 0$. It must be shown that $\psi(a) / \psi(b) = \psi(a / b)$. Now $\mu(a)$ and $\mu(b)$ are in T, so $\mu(a) / \mu(b) = \mu(a / b)$ by hypothesis 1). By hypothesis 2), $\tau(\mu(a) / \mu(b)) = \tau(\mu(a)) / \tau(\mu(b))$. But $\tau(\mu(a)) = \psi(a)$ and similarly for b by hypothesis 3), and $\tau(\mu(a) / \mu(b)) = \tau(\mu(a / b)) = \psi(a / b)$, so $\psi(a) / \psi(b) = \psi(a / b)$.

Thus, any such extension of Z would lead to a contradiction, because ψ does not preserve division.

Corollary 4.3

There is no extension T of the integers such that a ring homomorphism from T to C_m^r will preserve division.

By defining a new division operation for Z, it is possible to force ψ to preserve division. The fact that $\psi \circ \phi$ is the identity map on C_m^r will be used.

Definition 4.4

Let a and b be integers such that $\psi(a) / \psi(b)$ is defined in C_m^r. Then $a /_\psi b = \phi(\psi(a) / \psi(b))$.

This division operation is closely related to the original division of Z. It is the old operation with the operands reduced modulo r^m to values in S_m^r. This new division is preserved by ψ.

Theorem 4.5

$\psi(a /_\psi b) = \psi(a) / \psi(b)$.

Proof: This is clear, because $\psi \circ \phi$ is the identity on C_m^r.

Comparison to Modular Division

The divisions in Z and C_m^r are not inverse operations for multiplication. This is seen from the forms $A = Q \cdot B + R$ and $A = Q * B \oplus R$ in which the quotient $Q = A/B$ appears. Because of the remainder R, it is not in general possible to "un-do" a division by multiplying by the denominator; there are numbers A and B such that $(A/B)B \neq A$. In Z you have also the property for any integers A and B, with $B \neq 0$, that $(A \cdot B)/B = A$, while in C_m^r this will not be generally true, because in C_m^r the product is modular. In C_4^2, let $A = 5$ and $B = 3$. Then $5/3 = 1$, while $(5/3)*3 = 3$, so $(A/B)*B \neq A$. Also, $5*3 = -1$, so $(5*3)/3 = 0$ and $(A*B)/B \neq A$.

The fact that C_m^r and $Z_r m$ are isomorphic as rings brings up the question of how division in C_m^r relates to modular "division" in

$Z_r m$. Recall that a linear congruence $BX \equiv A \bmod p$ has a unique solution for X if and only if B and p are relatively prime. This unique value is taken to represent A/B in $Z_r m$. To avoid confusing the two division operations, the symbol % will be used to indicate the modular quotient.

The first significant difference between the divisions is that, when defined, modular division is the opposite of multiplication. If division by B is defined, then $A \% B$ satisfies $B(A \% B) \equiv A \bmod p$ because that is how the operation was defined. It is also true that $(AB) \% B = A$, because the number $(AB) \% B$ is the solution of the congruence $B \cdot X \equiv A \cdot B \bmod p$. Neither of these are true in C_m^r unless the division is exact and there is no overflow.

Supposing both quotients are defined, when do they agree and disagree? The C_m^r division A/B is defined for all pairs of numbers A and B with $B \neq 0$, while the modular division $A \% B$ of $Z_r m$ is defined only for those pairs where $(B, r^m) = 1$. If A/B is exact, if the remainder $R = 0$, then $B \cdot (A/B) = B^*(A/B) = A$, so $B \cdot (A/B) \equiv A \bmod r^m$. If $(B, r^m) = 1$ as well, then $A/B = A \% B$. On the other hand, if the division A/B is not exact, if $R \neq 0$, then $B \cdot (A/B) = B^*(A/B) \neq A$, and A/B does not satisfy the congruence $B \cdot X = A \bmod r^m$. Thus, $A/B \neq A \% B$ for those cases where $A \% B$ is defined. This proves the following.

Theorem 4.6

$A/B \equiv A \% B \bmod r^m$ if and only if $(B, r^m) = 1$ and A/B has remainder $R = 0$.

Another aspect of this difference is the lack of correspondence between the reciprocal (1/A or 1 % A) of a number A and its multiplicative inverse. In $Z_r m$, the congruence $AX \equiv 1 \bmod r^m$ has a solution if and only if A and r^m are relatively prime; the solution is unique, and is the reciprocal 1 % A. By its definition, 1 % A is the inverse of A. This will not be the case in C_m^r. Because of the isomorphism $\eta^*: C_m^r \to Z_r m$, the element A in C_m^r will have multiplicative inverse B if and only if $\eta^*(A)$ in $Z_r m$ has the inverse $\eta^*(B)$. However, if $|A| > 1$ and B is the inverse of A, then $1/A = 0$ and $|B| > 1$. Thus, for C_m^r, in only the trivial case of +1 or -1 will the reciprocal and multiplicative inverse of an element be the same.

This situation is observed in the 2's complement system C_m^2 used on computers. Each <u>odd</u> number A has a multiplicative inverse, because A and 2^m are relatively prime. An even number B has no inverse because $(B, 2^m) \neq 1$, meaning $BX \equiv 1 \bmod 2^m$ has no solution. The inverse of an odd number A is also odd, but if $|A| > 1$, then $1/A = 0$; zero is not the multiplicative inverse for any element of C_m^2. On the IBM/370, in C_{32}^2, the inverse of 3 is -1431655765, because 3 * (-1431655765) = 1, while the reciprocal 1/3 = 0.

The System $(C_m^r, /)$

Consider the system consisting of the numbers C_m^r together with one operation, division. It will be seen, and not surprisingly, that most of the classical algebraic axioms are not true for this system. The first to investigate is closure. Division was defined on C_m^r only for those pairs (A,B) where the integer result A/B represented

an element of C_m^r. This excluded division by zero, and, in an unbalanced system, the pair (Nm,-1). An attempt to reduce the system so division is defined for all pairs will not succeed. This would require that the element zero not be included, but zero is the result of any division where the denominator has larger magnitude than the numerator. Thus, the system is not closed for division.

To avoid this "closure" problem, it is possible to extend the definition of division. For those pairs (A,B) in C_m^r for which A/B is not defined, let the quotient Q = A with remainder R = 0. For convenience, the symbol A/B will be used for this quotient also. This gives closure under division, and is not without precedent; IBM/370 FORTRAN language, using C_{32}^2, operates this way. When a "divide exception" occurs and the computer recognizes a failure of its division algorithm, FORTRAN prints an error message and continues program execution. The unchanged dividend becomes the quotient with a remainder of zero. (The term "closure" is not quite valid here, because most FORTRAN systems will only allow this extension to be invoked a limited number of times.) In principle, this division could be used in most high-level programming languges; in practice, it is not.

The properties of associativity and commutativity do not generally hold for division operations. That is true here. Associativity is seen to fail when (4/3)/2 = 0 and 4/(3/2) = 4. Commutativity fails with 3/4 = 0 and 4/3 = 1.

Under the extended definition of division, there are two right identities for the system, the numbers 1 and 0. This is because X/1 = X and X/0 = X for any X in C_m^r. Zero is a right identity

only under the extended definition. There is, however, no left identity; for an element X with sufficiently large magnitude, and any other element A in C_m^r, the number A/X has smaller magnitude than X, so that $A/X \neq X$. Both left and right inverses exist for all elements X of C_m^r except $X = 0$, which has no right inverse. In fact, if $X \neq 0$, then X is its own inverse, since $X/X = 1$. $A = 1$ is the left inverse of $X = 0$, because $1/0 = 1$ under the extended definition. But $0/A = 0$ for each element A in C_m^r, so zero has no right inverse. Neither the left inverse nor right inverse is unique, in general. Because any element except zero is its own inverse, repeated division associated to the left yields zero, as $(X/X)/X = 0$ if $X \neq \pm 1$. The multiple quotient $(\ldots ((A/B)/C) \ldots /D)/E$ will be zero if $|A|$ is less than the magnitude of any other argument. Associating to the right gives $X/(X/X) = X$ and $X/(X/(X/X)) = 1$ for any X in C_m^r; a multiple quotient $X/(X/(\ \ldots\ (X/X)\ \ldots\ ))$ will be X or 1 depending on the number of arguments.

Division on C_m^r is distinct from addition and multiplication. It is not modular, and does not derive from primary division on the integers by means of a modular homomorphism. The next step in the study of division will consider its interaction with the other operations. First, however, the comparison relations on C_m^r will be investigated.

CHAPTER 5

THE COMPARISON RELATIONS

The ability to compare any two numbers, and to determine which, if either, is larger, is a vitally important characteristic of the integers. Occurring because the ring of integers is ordered, it is one of many significant properties deriving from the order on Z. The radix complement system, simulating a subset of the integers, would be of much less value if it did not reflect, at least in part, this order. Unfortunately, the modular nature of C_m^r precludes it having an order, so most order-related properties cannot hold. The comparability of numbers can, nevertheless, be retained.

The concept of an ordered ring generalizes that of an ordered field and is expressed in the following definitions.

Definition R.1

Let R be a ring. A non-empty subset P of R is called a positive class if it satisfies each of the following:

i) P is closed under addition in R,

ii) P is closed under multiplication in R, and

iii) each element X in R satisfies exactly one of the relations $X \in P$, $X = 0$, or $X \notin P \cup \{0\}$.

Definition R.2

A ring R is said to be ordered if it contains a positive class P. An element A of R which is in P is called positive, and this situation is denoted by the symbol $A > 0$ (or $0 < A$). If an element A is either positive or zero, the symbol $A \geq 0$ (or $0 \leq A$) is used. For elements A and B of R, if $A - B \in P$ then A is greater than B (B is less than A) and this is written as $A > B$ ($B < A$). If $A - B$ may also be zero, the symbol $A \geq B$ ($B \leq A$) is used.

Definition R.2 establishes the means for comparing integers. The characteristics of the positive class expressed in Definition R.1 then lead to the other order-related properties of Z, including the algebraic techniques needed to manipulate and solve equations of inequality.

The radix complement system C_m^r has both positive and negative integers among its signed value representatives. It is not, however, an ordered ring; it cannot satisfy Definition R.1 and has no positive class. This is because C_m^r is modular. Of the two elements 1 and -1, exactly one must be positive, the other non-positive. Since each of these elements additively generates the entire ring C_m^r, and a positive class is closed under addition, both of them would be positive, a contradiction.

The term "positive" might cause confusion. The radix complement system, in its usual signed value form, contains a subset of "positive" elements. This name, however, is applied only to distinguish

those elements of C_m^r which represent numbers from the positive class of Z. It does not imply the existence of a positive class in C_m^r. This usage is quite handy and will not be discarded.

C_m^r does not have a positive class, so Definition R.2 cannot be applied to provide relations for comparing elements. These relations can be derived indirectly by restricting the integer order relations $<$, $\leq$, $\geq$ and $>$ to S_m^r, the set of signed integer values for C_m^r. Produced in this way, they do not come from an order on C_m^r, and are thus not proper order relations. To emphasize this fact, they will be called comparison relations, or c-relations.

Definition 5.1

Let A and B be elements of C_m^r. The relation A .LT. B is true (.LE., .GE., .GT.) if and only if the relation $\phi(A) < \phi(B)$ is true ($\leq$, $\geq$, $>$, correspondingly), where ϕ is the signed value map from C_m^r to Z. The relations .LT., .LE., .GE., and .GT. are called the comparison relations (c-relations) on C_m^r. The relations .EQ. and .NE. are defined as the relations $=$ and $\neq$ on C_m^r, respectively.

The last two relations, although not c-relations, are included for consistency, because the entire group of six (.LT., .LE., .EQ., .GE., .GT., and .NE.) often appear in high level programming languages as relational operators. Other symbols may be used for them in various languages. The relation X .LE. Y is true if either X .LT. Y or X. EQ. Y is true. Similarly, X .GE. Y if either X .GT. Y or X .EQ. Y. These follow from Definition 5.1 and the order properties

on Z. Note also that X .LT. Y if and only if Y .GT. X and X .LE. Y if and only if Y .GE. X.

None of these relations provides an order for the ring C_m^r. However, two of them do form topological orderings on the set C_m^r. Topological orderings deal with elements of a system as abstract "points," ignoring any algebraic structure of the system. They are relations concerned with "maximal" and "minimal" elements of subsets within the system. The three to be considered are the partial order, the total (linear) order, and the "well-ordering."

A partial order Q on a set T is a relation on T having the properties of reflexivity (xQx for each x in T), anti-symmetry (xQy and yQx imply x = y for x and y in T), and transitivity (xQy and yQz imply xQz for x, y and z in T). If Q is a partial order on T, and has the property that either xQy or yQx for each x and y in T, then Q is a total (linear) order. The set T is well-ordered by Q if Q is a total order, and T has the property that each non-empty subset U of T has a "first point" under the order Q; by "first point" is meant a point x in U such that xQy for each y in U.

Theorem 5.2

1) .LE. is a partial order on C_m^r.

2) .LE. is a total order on C_m^r.

3) C_m^r is well-ordered by .LE. .

Proofs: 1) If A is in C_m^r then $\phi(A) \leq \phi(A)$, so A .LE. A and .LE. is reflexive. For A and B in C_m^r, if A .LE. B and

B .LE. A, then $\phi(A) \leq \phi(B)$ and $\phi(B) \leq \phi(A)$, so $\phi(A) = \phi(B)$. Since ϕ is one-to-one, this implies A = B, and so .LE. is anti-symmetric. With A, B and C in C_m^r, if A .LE. B and B .LE. C, then $\phi(A) \leq \phi(B)$ and $\phi(B) \leq \phi(C)$. But then $\phi(A) \leq \phi(C)$, so A .LE. C, and .LE. is transitive.

2) Part 1) shows .LE. is a partial order. If A and B are elements of C_m^r, then either $\phi(A) \leq \phi(B)$ or $\phi(B) \leq \phi(A)$ must hold. Because ϕ is one-to-one this means either A .LT. B or B .LT. A. Therefore, .LE. is a total order.

3) This follows because C_m^r is finite and has a total order.

Corollary 5.3

1) .GE. is a partial order on C_m^r.

2) .GE. is a total order on C_m^r.

3) C_m^r is well-ordered by .GE. .

The relations .LT. and .GT. have the transitive property, but fail to exhibit reflexivity or anti-symmetry.

Computer Evaluation of C-Relations

It is important to be able to evaluate the comparison relations using the radix complement system and its operations. Definition R.2 specified the order relations on Z by saying $x > y$ if and only if $x - y > 0$. Since the sign of an element in C_m^r may be tested, this suggests using subtraction to establish the c-relations. This will not always work, unless certain other constraints are met. Before investigating those conditions, some further information concerning

the homomorphism ψ: $Z \rightarrow C_m^r$, the signed value map ϕ: $C_m^r \rightarrow Z$, and the c-relations may prove useful. In what follows, no distinction will be made between elements of C_m^r and their corresponding integer signed values, unless clarity demands. Use of an integer order relation implies its arguments should be taken as signed integer values.

Lemma 5.4

If x is an integer such that $Nm \leq x \leq Pm$, then $\phi(\psi(x)) = x$.

Proof: This should be clear from the definitions of ϕ and ψ.

Lemma 5.5

If x and y are integers such that $Nm \leq x < y \leq Pm$, then $\psi(x)$ .LT. $\psi(y)$. The same is true if $<$ is replaced by any other order relation on Z, and .LT. by the corresponding c-relation on C_m^r.

Proof: By contradiction. Suppose $\psi(x)$ .GE. $\psi(y)$. Then, by definition of .GE., $\phi(\psi(x)) \geq \phi(\psi(y))$. Since $N_m \leq x \leq P_m$ and $N_m \leq y \leq P_m$, the preceeding lemma gives $x \geq y$, a contradiction. The proof for the other relation-comparison relation pairs is similar.

Corollary 5.6

Let x be an integer. If $0 < x \leq Pm$ then $\psi(x)$ .GT. 0, and if $Nm \leq x < 0$ then $\psi(x)$ .LT. 0. (And similarly for the other relation-comparison relation pairs.)

Proof: Note that ψ is a homomorphism, so $\psi(0) = 0$.

Lemma 5.7

1) If x is an integer with $Pm < x \leq Pm - Nm$, then $\psi(x)$ .LT. 0.

2) If x is an integer with $Nm - Pm \leq x < Nm$, then $\psi(x)$ .GT. 0.

Proofs: 1) This is clear because C_m^r is a modular system isomorphic to $Z_r m$, and Nm in some sense "follows" Pm, as $Pm \oplus 1 = Nm$. Thus, $\psi(Pm + 1) = Nm$, $\psi(Pm + 2) = Nm \oplus 1$, ... , $\psi(Pm - Nm) = -1$.

2) The argument here is similar, except Pm "preceeds" Nm, as $Nm \ominus 1 = Pm$. Hence, $\psi(Nm - 1) = Pm$, $\psi(Nm - 2) = Pm \ominus 1$, ... , $\psi(Nm - Pm) = 1$.

The Subtraction Test

To see how important this is to computer users, we examine the subtraction test. Subtraction alone will not always correctly determine the c-relations between two elements of C_m^r. If $X \ominus Y$ overflows, it will have the opposite sign from that required for a valid comparison of X and Y. In C_4^2, the subtraction gives $(-7) \ominus 6 = 3$, indicating -7 .GT. 3, which is false.

Theorem 5.8

Let X and Y be elements of C_m^r. If $X \ominus Y$ does not overflow, then X .GT. Y if and only if $(X \ominus Y)$ .GT. 0. (The same is true if .GT. is replaced by .GE. .)

Proof: If X .GT. Y, then $X > Y$ and $0 < X - Y$. Since $X \ominus Y$ does not overflow, $X - Y \leq Pm$, so $0 < X - Y \leq Pm$, and by Lemma 5.6, $\psi(X - Y)$ .GT. 0. Since $\psi(X - Y) = X \ominus Y$, this gives $(X \ominus Y)$ .GT. 0.

If $(X \ominus Y)$ .GT. 0, then $\phi(X \ominus Y) > 0$. Since $X \ominus Y$ does not overflow, $\phi(X \ominus Y) = X - Y$, so $X - Y > 0$ and $X > Y$. Thus X .GT. Y.

A similar proof holds for the .GE. c-relation.

Theorem 5.9

Let X and Y be elements of C_m^r. If X ⊖ Y overflows, then X .GT. Y if and only if (X ⊖ Y) .LT. 0.

Proof: If X .GT. Y, then 0 < X - Y. Then, since X ⊖ Y overflows, Pm < X - Y ≤ Pm - Nm. Lemma 5.7 now shows that ψ(X - Y) .LT. 0, and this gives (X ⊖ Y) .LT. 0.

If (X ⊖ Y) .LT. 0, then ϕ(X ⊖ Y) < 0. The overflow condition and bounds on subtraction imply Nm - Pm ≤ X - Y < Nm or Pm < X - Y ≤ Pm - Nm. If Nm - Pm ≤ X - Y < Nm, then Lemma 5.7 gives (X ⊖ Y) .GT. 0, a contradiction. Thus, Pm < X - Y ≤ Pm - Nm, so 0 < X - Y. Therefore, X > Y and X .GT. Y.

It is not necessary to extend Theorem 5.9 to the .GE. c-relation. If X ⊖ Y overflows, then X .NE. Y.

This last theorem specifies exactly when the subtraction test fails. It fails when the subtraction overflows. Thus, to have a better idea of when the test succeeds, it will be useful to characterize overflow for subtraction.

Theorem 5.10

Suppose X and Y are elements of C_m^r, such that X .GT. Y. Then X ⊖ Y does not overflow if any of the following occur:

1) X and Y have the same sign,

2) Y .GE. (X ⊖ Pm), or

3) Y .LT. (X ⊖ Pm) and (X ⊖ Pm) .GE. X.

The subtraction X ⊖ Y will overflow if

4) Y .LT. (X ⊖ PM) and (X ⊖ Pm) .LT. X.

Conditions 1), 2) and 3) are not mutually exclusive. The first is a partial combination of the other two. It is included because of the simple relation between X and Y. Conditions 2), 3) and 4) are both exhaustive and mutually disjoint.

The next two lemmas will be useful in the proof of Theorem 5.10.

Lemma 5.11

Suppose X and Y are elements of C_m^r, with Y .GT. 0. Then (X ⊖ Y) .LT. X if and only if $Nm \leq X - Y < Pm$.

Proof: Suppose (X ⊖ Y) .LT. X. If $X - Y < Nm$, then $Nm - Pm \leq X - Y < Nm$. Lemma 5.7 then implies $\psi(X - Y)$ .GT. 0, so (X ⊖ Y) .GT. 0. It also follows that $X < Nm + Y$, and, because $Nm + Y \leq 1$ for either a balanced or unbalanced system, that $X \leq 0$ and X .LE. 0. The transitive property of .LT. then implies X .LT. (X ⊖ Y). This contradiction means $Nm \leq X - Y$. Since $Y > 0$, it follows that $X - Y < X$, so $X - Y < Pm$, and therefore, $Nm \leq X - Y < Pm$.

Now suppose $Nm \leq X - Y < Pm$. As before, $X - Y < X$, so $Nm \leq X - Y < X \leq Pm$. Lemma 5.5 then implies $\psi(X - Y)$ .LT. X, so (X ⊖ Y) .LT. X.

Lemma 5.12

If X and Y are elements of C_m^r, with Y .GT. 0 and X .LE. (X ⊖ Y), then $X - Y < Nm$.

Proof: This is essentially the contrapositive of one part of Lemma 5.11. X .LE. (X ⊖ Y) implies that $Nm \leq X - Y < Pm$ is not true. As in Lemma 5.11, $X - Y < Pm$, so it must be true that $X - Y < Nm$.

Proof of Theorem 5.10:

1) In this case, either 0 .LT. Y .LT. X .LE. Pm or Nm .LE. Y .LT. X .LT. 0. Thus, either $0 < Y < X \leq Pm$ or $Nm \leq Y < X < 0$. The range restrictions in both cases guarantee $Nm \leq X - Y \leq Pm$, for either a balanced or unbalanced system, so there is no overflow in $X \ominus Y$. The condition X .GT. Y is not needed here.

2) If X and Y have the same sign, case 1) provides the conclusion. Suppose X and Y do not have the same sign. Then, since $X > Y$, it must be true that $X \geq 0$ and $Y \leq 0$. Thus $X - Y \geq 0$, and $Nm \leq X - Y$. Now, with X .GT. Y and Y .GE. $(X \ominus Pm)$, it follows that X .GT. $(X \ominus Pm)$ or $(X \ominus Pm)$ .LT. X. Because Pm .GT. 0, Lemma 5.11 gives $Nm \leq X - Pm < Pm$. This means $X \ominus Pm$ does not overflow, so $\phi(X \ominus Pm) = X - Pm$. Then $Y \geq \phi(X \ominus Pm)$, so $Y \geq X - Pm$ and $X - Y \leq Pm$. Therefore, $Nm \leq X - Y \leq Pm$, and $X \ominus Y$ does not overflow.

3) As in part 2), it is only necessary to consider when X and Y have opposite signs, with $Nm \leq X - Y$. For the other bound on $X - Y$, the hypothesis gives X .LE. $(X \ominus Pm)$, so applying Lemma 5.12 yields $X - Pm < Nm$. Since $Nm \leq Y$, this becomes $X - Pm < Y$, and $X - Y < Pm$. Thus $Nm \leq X - Y \leq Pm$, so $X \ominus Y$ does not overflow.

4) Here, $Y < \phi(X \ominus Pm)$. As in part 2), $(X \ominus Pm)$ .LT. X implies that $\phi(X \ominus Pm) = X - Pm$, so $Y < X - Pm$. Thus, $Pm < X - Y$ and $X \ominus Y$ overflows. The overflow is positive because X .GT. Y.

Part 4) of Theorem 5.10 shows that precisely when Y .LT. $(X \ominus Pm)$ and $(X \ominus Pm)$ .LT. X, the subtraction test will

yield (X ⊖ Y) .LT. 0 when Y .LT. X. This is exactly the opposite of what one would hope to have; the subtraction test fails to provide the proper c-relation. The condition under which this will occur may be loosely stated as being when X and Y are "too far apart" in the "positive" direction.

The significance of the failure of subtraction to correctly determine the comparison relations lies in the fact that in some major computer languages and on some computers, particularly the smaller mini- and micro-computers, users frequently rely on subtraction to provide these comparisons. In the larger computers, where the complexity of arithmetic logic circuitry is less a factor, there is often a single machine instruction which will make valid algebraic comparisons between any two elements of C_m^r. This comparison may actually use a subtraction process, but extra logic circuitry is provided to compensate when overflow occurs.

As an example, consider the IBM/360 and 370 computers. On these machines, single instruction comparisons are available to provide the correct c-relations between any two elements in C_{32}^2 or C_{16}^2. The FORTRAN IV language on these machines makes use of this compare instruction in the "logical IF" statement to permit testing the c-relations. The "relational operators" .GT., .GE., .EQ., .LE., .LT. and .NE. provide valid comparisons between any two 2's complement numbers. FORTRAN also permits comparisons to be attempted indirectly using the less reliable "arithmetic IF" statement. This statement does not actually test the c-relations. Instead, it tests a number, perhaps the result of evaluating an arithmetic expression, to see if it is negative, zero

or positive. A test of c-relations is implemented by performing a subtraction and testing the result. As shown in Theorem 5.10, this procedure using the arithmetic IF can fail to provide the correct c-relation. The logical IF will never fail. This situation may be seen in the two programs below, with the output from their executions.

Example 5.13

This program uses the logical IF statement.

```
 10   C   EXAMPLE USING LOGICAL  'IF'
 20   C
 30           IA =  1500000000
 40           IB = -1500000000
 50           IF (IA .GT. IB)  GO TO 10
 60           WRITE(6,100)  IB,IA
 70  100      FORMAT(//////1X,I12,'  IS GREATER THAN ', I12)
 80           STOP
 90   10      WRITE(6,100)  IA,IB
100           STOP
110           END
END OF DATA

RUN
  1500000000  IS GREATER THAN  -1500000000
EDIT
```

Example 5.14

This program attempts to compare the same two numbers using the arithmetic IF statement.

```
 10   C    EXAMPLE USING ARITHMETIC 'IF'
 20   C
 30           IA =   1500000000
 40           IB =  -1500000000
 50           IF(IA - IB) 10,20,30
 60   10      WRITE(6,100) IB,IA
 70   100     FORMAT(//////1X,I12,' IS GREATER THAN ', I12)
 80           STOP
 90   20      WRITE(6,101) IA,IB
100   101     FORMAT(1X,I12,' EQUALS ',I12)
110           STOP
120   30      WRITE(6,100) IA,IB
130           STOP
140           END
END OF DATA

RUN
 -1500000000  IS GREATER THAN  1500000000
EDIT
```

Many FORTRAN IV programmers use the arithmetic IF for comparisons, even when the logical IF is available. Older versions of FORTRAN or versions for smaller computers may not offer the logical IF statement. In such languages, no single statement can perform valid comparisons for all pairs of radix complement numbers. To be fair, computer manufacturers have never claimed that comparison by subtraction with the arithmetic IF gave valid results with radix complement numbers. Some early versions of FORTRAN would not even permit such usage; the arithmetic IF could not be applied to elements of C_m^r, but only to floating point numbers.

The situation with mini- and micro-computers is different. These small machines have rather severe restrictions on the complexity of their arithmetic circuitry. As a result, they are not usually able to implement the valid single operation "compare" of larger machines. Many of these do, nevertheless, have a "compare" operation; it con-

sists of performing a subtraction of the two arguments (radix complement integers) and setting a collection of indicators or "flags" according to the result. The flags set usually include "result positive," "result negative," "result zero," and "overflow." The latter is very important because it indicates if the subtraction comparison is in the failure situation.

Avoiding Subtraction Error

The subtraction error in radix complement comparisons is fairly subtle. It can only occur when numbers of opposite sign are compared, and not always then. There are several ways it can be overcome.

The simplest approach is probably to avoid it. If the numbers to be compared all have the same sign, the subtraction test will not fail. Similarly, if magnitudes of the numbers are restricted so that subtraction overflow cannot happen, then this test will suffice. If the C_m^r system has a great many elements, restriction may not prove a hardship. In C_{32}^2 on the IBM/370, integer values run from -2147483648 ($-2^{31} = N_{32}^2$) through 2147483647 ($2^{31} - 1 = p_{32}^2$). If the numbers to be compared always lie between -1073741824 (-2^{30}) and 1073741823 ($2^{30} - 1$), then subtraction will not overflow and may be used for comparisons. However, for C_{16}^2 on the IBM 1130, a valid comparison would require restriction to numbers in the range from -16384 (-2^{14}) to 16383 ($2^{14} - 1$). This is a severe limitation.

Magnitude and sign restrictions may be feasible in particular situations, but unacceptable in more generalized cases, such as in

high-level programming languages. To have comparison by subtraction valid for arbitrary pairs in C_m^r, it must be done in two steps. One possibility is to first check signs of the values being compared. The second step is then subtraction, but need be performed only if the signs checked were the same. The subtraction will yield a valid comparison. Alternatively, the subtraction could be done first, followed by a check for overflow. The presence of overflow demands reversal of the c-relation indicated by the subtraction; .LT. (.LE.) for .GT. (.GE.) or vice-versa. The choice between these schemes is hardware dependent, and rests on the relative ease of comparing signs or checking for overflow.

Another way to avoid the subtraction error is to imbed the C_m^r system in the C_k^r system for the comparison subtraction, where $k > m$. This is a relatively simple procedure if the logic circuitry is so designed; $k - m$ extra digits are added as the most significant digits to each number being compared. The new digits added to a positive number are all zero, and all $r-1$ for a negative number. The signed integer value is thus unchanged for an element imbedded in this fashion. Furthermore, subtraction in C_k^r of two elements from C_m^r will not overflow; elements from C_m^r may be validly compared by subtraction in C_k^r. Computers allowing double precision integer arithmetic may take $k = 2m$. This technique could be used for built-in comparison operations; one extra digit $(k = m+1)$ is sufficient.

In what follows, the comparison relations will be assumed as part of C_m^r; how they are determined will not matter. Because the c-relations do not arise from an order, many properties of the order

relations on Z will not hold for C_m^r. The algebra of inequalities in Z is, however, quite useful, and it would be interesting to see how much of it, if any, can be carried over to C_m^r.

CHAPTER 6

INEQUALITIES WITH C-RELATIONS

The algebra of inequalities provided by the order relations on Z allows the manipulation and solution of equations of inequality. Included are such theorems as these. (In Theorems I.1, I.2, and I.3, x, y and a represent integers from Z.)

Theorem I.1

If $x < y$, then $x + a < y + a$.

Theorem I.2

If $x < y$, and $a > 0$, then $a \cdot x < a \cdot y$.

If $x < y$, and $a < 0$, then $a \cdot x > a \cdot y$.

Theorem I.3

If $x < y$, and $a > 0$, then $x/a \leq y/a$.

If $x < y$, and $a < 0$, then $x/a \geq y/a$.

The division is the primary quotient in Z.

The first two of these will only partly carry over to C_m^r, and that with some difficulty. The problem, illustrated by Theorem 5.9, arises when overflow occurs. Theorem I.3 has an unusual form (with $\leq$) because the division is the primary quotient, rather than the rational quotient. Overflow does not occur in division, and this theorem carries over,

mutatis mutandis to C_m^r. Inequalities using the c-relations will be called c-inequalities.

The Modulo Reduction Factor

Since overflow will play an important role with c-inequalities, further consideration of it may prove helpful.

Lemma 6.1

Let X and Y be elements of C_m^r, and let $@_c$ be any of the arithmetic operations in C_m^r, with $@_z$ the corresponding operation in the integers. Then $\phi(X @_c Y) = \phi(X) @_z \phi(Y) + kr^m$, for some integer k.

This just expresses the modular nature of arithmetic in C_m^r, as in Definition 2.10. Again, the distinction between X and $\phi(X)$ is often ignored; the equation in Lemma 6.1 then appears as $\phi(X @_c Y) = X @_z Y + kr^m$. The integer k provides information concerning overflow, and will be called the modulo reduction factor (m.r.f.).

Lemma 6.2

1) There is overflow if and only if $k \neq 0$.

2) There is positive overflow if and only if $k < 0$.

3) There is negative overflow if and only if $k > 0$.

Proof: 1) If there is overflow, then $\phi(X @_c Y) \neq X @_z Y$, so it must be true that $kr^m \neq 0$. Thus $k \neq 0$. If $k \neq 0$, then $\phi(X @_c Y) \neq X @_z Y$, and there is overflow.

2) If there is positive overflow, then $X @_z Y > Pm$, and $-(X @_z Y) < -Pm$. Since $Nm \leq \phi(X @_c Y) \leq Pm$, it follows that $X @_z Y + kr^m \leq Pm$. Then $kr^m \leq Pm - (X @_z Y) < 0$, so $k < 0$.

On the other hand, if $k < 0$, the bounds on $\phi(X \circledcirc_c Y)$ force $X \circledcirc_z Y > Pm$, and there is positive overflow.

3) The proof here is similar to that in 2).

For the operation of addition, it is possible to give values for the modulo reduction factor k.

Lemma 6.3

Let X and Y be elements of C_m^r. Then $X \oplus Y$ gives positive overflow if and only if $\phi(X \oplus Y) = X + Y - r^m$ $(k = -1)$, and negative overflow if and only if $\phi(X \oplus Y) = X + Y + r^m$ $(k = 1)$.

Proof: Because $Nm \leq X \leq Pm$ and $Nm \leq Y \leq Pm$, we have that $2 \cdot Nm \leq X + Y \leq 2 \cdot Pm$. Since $Pm - Nm + 1 = r^m$, this implies that $-r^m \leq X + Y \leq r^m$. This in turn, with the bounds on $\phi(X \oplus Y)$, forces $|k| \leq 1$. Application of the previous Lemma 6.2 completes the proof.

The modulo reduction factor with multiplication depends more on the operands, and may have large value. It is closely related to the digits truncated (or not formed) when the C_m^r product is produced. To see this, a more general definition of the modulo reduction factor will be introduced. Mappings used are those of Chapter 2: the signed value map $\phi: C_m^r \to Z$, the unsigned value map $\eta: C_m^r \to Z$, and the natural homomorphism $\psi: Z \to C_m^r$.

Definition 6.4

Let a be an integer, so that $\psi(a)$ is in C_m^r. Then $\phi(\psi(a)) = a + kr^m$, and k is the modulo reduction factor.

The definition of the modulo reduction factor in Lemma 6.1 is

a special case of this. If A and B are elements of C_m^r, then $\phi(A)$ and $\phi(B)$ are integers. Let $@_c$ and $@_z$ be corresponding operations in C_m^r and Z, respectively. Taking $a = \phi(A)\ @_z\ \phi(B)$ gives $\psi(a) = A\ @_c\ B$, because $\psi \circ \phi$ is the identity map. The expression for k then becomes $\phi(A\ @_c\ B) = \phi(A)\ @_z\ \phi(B) + kr^m$, as in Lemma 6.1.

Many computers, including the IBM/360 and 370, form a double precision result for integer multiplication. With operands in C_m^r, the product is produced in C_{2m}^r; truncating the m most significant digits of the extended result gives the product in C_m^r. The digits truncated, taken as an element of C_m^r, will help provide the modulo reduction factor for the multiplication. Signed and unsigned values from both C_m^r and C_{2m}^r will be needed. These will be provided by ϕ and η, each applied to both systems.

Let A be an element of C_{2m}^r, with digital form

$$a_{2m-1}a_{2m-2} \cdots a_{m+1}a_m a_{m-1} \cdots a_1 a_0.$$

The digits $a_{2m-1}a_{2m-2} \cdots a_{m+1}a_m$ are the m most significant digits. They can be taken as an element of C_m^r; this element will be represented by A_u. The digits $a_{m-1} \cdots a_1 a_0$ also represent an element A_1 in C_m^r. If A is the machine product of two elements from C_m^r, then A_u is the part truncated to give the radix complement result A_1.

Lemma 6.5

Suppose A is an element of C_{2m}^r, with most and least significant parts A_u and A_1 as described. Then $\phi(A_1) = \phi(A) + kr^m$,

where

$$k = \begin{cases} -\eta(A_u) & \text{if } \phi(A_1) \geq 0 \text{ and } \phi(A) \geq 0, \\ -\eta(A_u) + r^m & \text{if } \phi(A_1) \geq 0 \text{ and } \phi(A) < 0, \\ -\eta(A_u) - 1 & \text{if } \phi(A_1) < 0 \text{ and } \phi(A) \geq 0, \text{ or} \\ -\eta(A_u) - 1 + r^m & \text{if } \phi(A_1) < 0 \text{ and } \phi(A) < 0. \end{cases}$$

Proof: $\psi(\phi(A)) = A_1$ so k is the modulo reduction factor of Definition 6.4, with $\phi(A_1) = \phi(A) + kr^m$. From Definition 2.7, $\phi(X) = \eta(X)$ if $\phi(X) \geq 0$, and $\phi(X) = \eta(X) - r^m$ if $\phi(X) < 0$. This may be applied to the equation for k in four cases.

Case 1. Suppose $\phi(A_1) = \eta(A_1)$ and $\phi(A) = \eta(A)$. Then $\eta(A_1) = \eta(A) + kr^m$, so $kr^m = -\eta(A) + \eta(A_1)$. Since $\eta(A) = \eta(A_u)r^m + \eta(A_1)$, this gives $kr^m = -\eta(A_u)r^m$, so $k = -\eta(A_u)$.

Case 2. $\phi(A_1) = \eta(A_1)$ and $\phi(A) = \eta(A) - r^{2m}$. Here, $kr^m = -\eta(A_u)r^m + r^{2m}$, so $k = -\eta(A_u) + r^m$.

Case 3. $\phi(A_1) = \eta(A_1) - r^m$ and $\phi(A) = \eta(A)$. Then $kr^m = -\eta(A_u)r^m - r^m$ and $k = -\eta(A_u) - 1$.

Case 4. $\phi(A_1) = \eta(A_1) - r^m$ and $\phi(A) = \eta(A) - r^{2m}$. Then $k = -\eta(A_u) - 1 + r^m$.

Lemma 6.5 gives the value of k in terms of the unsigned value $\eta(A_u)$. Ordinarily, the unsigned value is not accessible in a high-level programming language. Thus, it may be better to express k using the signed value $\phi(A_u)$. The next theorem does this.

Theorem 6.6

Let A, A_u and A_1 be given as in Lemma 6.5. Then

$\phi(A_u) = \phi(A) + kr^m$, where k is the modulo reduction factor.

If A_u and A have the same sign, or A_u is zero, then

$$k = \begin{cases} -\phi(A_u) & \text{if } \phi(A_l) \geq 0, \text{ or} \\ -\phi(A_u) - 1 & \text{if } \phi(A_l) < 0. \end{cases}$$

If A_u and A have opposite signs, then

$$k = \begin{cases} -\phi(A_u) + s(A_u) \cdot r^m & \text{if } \phi(A_l) \geq 0, \text{ or} \\ -\phi(A_u) - 1 + s(A_u) \cdot r^m & \text{if } (A_l) < 0. \end{cases}$$

Proof: The sign function $s: C_m^r \to \{1, 0, -1\}$. The expressions for k are obtained from Lemma 6.5 by substituting $\phi(A_u)$ or $\phi(A_u) + r^m$ for $\eta(A_u)$, depending on whether A_u is non-negative or negative, respectively.

The form of this theorem makes the next corollary obvious. In an unbalanced radix complement system, the most significant digit determines the sign of a number. This means the signs of A and A_u will be the same.

Corollary 6.7

Let C_m^r be unbalanced (r even), and let A, A_u and A_l be given as before. Then the modulo reduction factor k is given by

$$k = \begin{cases} -\phi(A_u) & \text{if } \phi(A_l) \geq 0, \text{ or} \\ -\phi(A_u) - 1 & \text{if } \phi(A_l) < 0. \end{cases}$$

Most significant radices in use are even, and the modulo reduction factor is closely related to the digits truncated when a double precision result is reduced to C_m^r. The situation is more complicated for odd

radix. Both parts of Theorem 6.6 must be used. This is illustrated in the following example.

Example 6.8

Consider the four digit 3's complement system, C_4^3. It is balanced, and elements appear as four digit numbers in base 3. Let $A = 1000$. Then $A_u = 10$ in C_2^3. Since $\phi(1000) = 27$ and $\phi(10) = 3$, the numbers A and A_u have the same sign. If $A = 1120$, then $A_u = 11$ in C_2^3. Now, however, $\phi(1120) = -39$ and $\phi(11) = 4$, so A and A_u have opposite signs. Note that $P_4^3 = 1111$ and $N_4^3 = 1112$ while $P_2^4 = 11$ and $N_2^4 = 12$.

Applying Theorem 6.6 to obtain the modulo reduction factor for a product $X*Y$ of elements from C_m^r requires that $A = \psi\{\phi(X)\cdot\phi(Y)\}$, where $\psi: Z \to C_{2m}^r$. Then A is the double precision machine product of X and Y.

C-Inequalities with Addition and Multiplication

The next lemma provides some insight into a relationship between positive and negative overflow. Because C_m^r is a ring, it is true that $(X \oplus Y) \oplus (-Y) = X$. The lemma indicates how overflow comes into play in the intermediate results as $(X \oplus Y) \oplus (-Y)$ is calculated.

Lemma 6.9

Let X and Y be elements of C_m^r, such that $Y \neq -Y$.

If $X \oplus Y$ $\left\{\begin{array}{l}\text{does not overflow,}\\ \text{has positive overflow,}\\ \text{has negative overflow,}\end{array}\right\}$ then

$(X \oplus Y) \oplus (-Y)$ $\left\{\begin{array}{l}\text{does not overflow,}\\ \text{has negative overflow,}\\ \text{has positive overflow.}\end{array}\right\}$

Proof: If $X \oplus Y$ does not overflow, then $\phi(X \oplus Y) = X + Y$. The restriction on Y implies $\phi(-Y) = -Y$. Thus there is an integer k, with $|k| \leq 1$, such that

$$\phi[(X \oplus Y) \oplus (-Y)] = \phi(X \oplus Y) + \phi(-Y) + kr^m = X + Y - Y + kr^m = X + kr^m.$$

Since $\phi[(X \oplus Y) \oplus (-Y)] = \phi(X) = X$, this means that $k = 0$, so $(X \oplus Y) \oplus (-Y)$ does not overflow.

Suppose $X \oplus Y$ has positive overflow. Then $\phi(X \oplus Y) = X + Y - r^m$. As before, there is some integer k, with $|k| \leq 1$, such that

$$X = \phi[(X \oplus Y) \oplus (-Y)] = (X + Y - r^m) - Y + kr^m = X - r^m + kr^m.$$

This means $k = 1$, so $(X \oplus Y) \oplus (-Y)$ has negative overflow. The case where $X \oplus Y$ has negative overflow is similar.

The next theorem establishes a basic property for c-inequalities involving addition.

Theorem 6.10

Let X, Y and A be elements of C_m^r.

1) If neither $X \oplus A$ nor $Y \oplus A$ overflow, or both do, then X .LT. Y if and only if $(X \oplus A)$ .LT. $(Y \oplus A)$.

2) If exactly one of $X \oplus A$ or $Y \oplus A$ overflows, then X .LT. Y if and only if $(X \oplus A)$ .GT. $(Y \oplus A)$.

Proof: We begin with two observations. First, X .LT. Y means that $X < Y$. Second, to have positive overflow from a sum requires both summands to be positive, while negative overflow requires both to be negative.

Proof of 1): Suppose that X .LT. Y. Then $X < Y$ and $X + A < Y + A$. If neither $X \oplus A$ nor $Y \oplus A$ overflow, then $Nm \leq X + A < Y + A \leq Pm$, and Lemma 5.5 gives $(X \oplus A)$ .LT. $(Y \oplus A)$. If both $X \oplus A$ and $Y \oplus A$ overflow, then they must both have positive overflow or both have negative overflow. This is seen by the remark above; if $X \oplus A$ has positive (negative) overflow, then A must be positive (negative), implying that $Y \oplus A$ must also have positive (negative) overflow. Suppose they both have positive overflow. Then $\phi(X \oplus A) = X + A - r^m$ and $\phi(Y \oplus A) = Y + A - r^m$. However, $X + A - r^m < Y + A - r^m$, so $(X \oplus A)$ .LT. $(Y \oplus A)$. The argument for negative overflow is similar.

Conversely, suppose $(X \oplus A)$ .LT. $(Y \oplus A)$, so $\phi(X \oplus A) < \phi(Y \oplus A.)$ If neither $X \oplus A$ nor $Y \oplus A$ overflow, this implies $X + A < Y + A$. Thus $X < Y$ and X .LT. Y. On the other hand, if both $X \oplus A$ and $Y \oplus A$ have positive overflow, then $\phi(X \oplus A) = X + A - r^m$ and $\phi(Y \oplus A) = Y + A - r^m$. This again gives $X < Y$, so X .LT. Y. As before, the argument with negative overflow is similar.

Before beginning the proof of the second statement in the theorem, another fact should be noted. If B, C and D are elements of C_m^r, with B .LT. C, such that exactly one of $B \oplus D$ and $C \oplus D$ overflow, then either $C \oplus D$ has positive overflow, or $B \oplus D$ has negative overflow. This is easily seen, because B .LT. C implies that $B + D < C + D$; to have exactly one term overflow, either $C + D > Pm$ or $B + D < Nm$.

Proof of 2): Suppose X .LT. Y. As noted above, if only one of

$X \oplus A$ and $Y \oplus A$ overflow, then either $X \oplus A$ has negative overflow, or $Y \oplus A$ has positive overflow. Suppose $X \oplus A$ has negative overflow. This means $\phi(X \oplus A) = X + A + r^m$. Since $Y \oplus A$ does not overflow, we have $\phi(Y \oplus A) = Y + A$. It must be shown that $X + A + r^m > Y + A$. With $X < Y$, bounds on X and Y give $Y - X \leq Pm - Nm < r^m$. Thus, $X + r^m > Y$, and so $X + A + r^m > Y + A$. Therefore, $(X \oplus A)$ .GT. $(Y \oplus A)$.

Suppose, instead, that $Y \oplus A$ has positive overflow. Then $\phi(X \oplus A) = X + A$ and $\phi(Y \oplus A) = Y + A - r^m$, so it is necessary to prove $X + A > Y + A - r^m$. The argument above established that $Y - X < r^m$. Thus, $X > Y - r^m$, so $X + A > Y + A - r^m$, and again $(X \oplus A)$ .GT. $(Y \oplus A)$.

For the converse in 2), assume $(X \oplus A)$ .GT. $(Y \oplus A)$. We will then suppose that X .GE. Y and attempt to produce a contradiction. If X .EQ. Y, then $(X \oplus A)$ .EQ. $(Y \oplus A)$, which clearly cannot be true. Thus, X .GT. Y, which is equivalent to Y .LT. X. Applying the first half of the proof for part 2) to this gives $(Y \oplus A)$ .GT. $(X \oplus A)$. This is equivalent to $(X \oplus A)$ .LT. $(Y \oplus A)$, and also clearly false. Therefore, it must be true that X .LT. Y.

This concludes the proof of Theorem 6.10.

The preceding theorem characterizes the preservation of the c-relation .LT. under addition. Similar statements may be made for the other c-relations. The question should now be raised as to whether a similar result will hold for the operation of multiplication. Unfortunately, if overflow is present, no such simple statement may be made. Without overflow, the situation is like that in the integers Z.

Theorem 6.11

Let X, Y and A be elements of C_m^r. If neither X*A nor Y*A overflow, then the following are true:

1) If A .GT. 0 then X .LT. Y if and only if (X*A) .LT. (Y*A),

2) If A .LT. 0 then X .LT. Y if and only if (X*A) .GT. (Y*A).

Proof: These both follow directly from the condition that there be no overflow. Because of this $\phi(X*A) = X \cdot A$ and $\phi(Y*A) = Y \cdot A$, so the c-relation .LT. is equivalent to the relation < in the integers.

With overflow in X*A or Y*A, or both, there is no simple solution. Success was achieved for addition because the amount of overflow was strictly limited; the modulo reduction factor had magnitude less than two. The m.r.f. in X*A or Y*A is not so restricted. It depends on both the signs and magnitudes of the operands. An example will illustrate this.

Example 6.12

Consider the 2's complement system with four digits (bits). In signed integer form, $C_4^2 = \{-8, -7, -6, \ldots, -1, 0, 1, 2, \ldots, 6, 7\}$. The modulo reduction factor k is determined from $\phi(X*Y) = X \cdot Y + kr^m$, where $\phi: C_m^r \to Z$ is the signed value map. In this system, it is true that 3 .LT. 7, and the c-relation between 3*A and 7*A is shown in the following table for various values of A.

A	3*A	7*A	c-relation of 3*A and 7*A	modulo reduction factors of 3*A	of 7*A
1	3	7	.LT.	0	0
3	-7	5	.LT.	-1	-1
4	-4	-4	.EQ.	-1	-2
5	-1	3	.LT.	-1	-2
6	2	-6	.GT.	-1	-3
7	5	1	.GT.	-1	-3

Depending on the magnitude of A, A .GT. 0, any c-relation is possible between 3*A and 7*A.

C-Inequalities with Division

Division on C_m^r is not a modular operation, and does not produce overflow. Except for the undefined quotient Nm/(-1) in an unbalanced system, it is the same as the primary quotient in Z.

Theorem 6.13

Let X, Y and A be elements of C_m^r, such that both X/A and Y/A are defined, with A .GT. 0. If X .LT. Y then (X/A) .LE. (Y/A), and the latter c-inequality cannot be more strict.

Proof: Again, no distinction will be made between an element B of C_m^r and its signed value $\phi(B)$. If X and Y have opposite signs, then X/A and Y/A have opposite signs, unless one or both of them are zero. Since A .GT. 0, the quotients have the same sign as their respective numerators, so (X/A) .LE. (Y/A). If either of X or Y are zero, take them as having opposite signs.

Consider the situation when X and Y have the same sign, letting R_1 and R_2 be the remainders from X/A and Y/A, respectively. From Chapter 3, X/A satisfies the integer equation $X = \phi(X/A) \cdot A + R_1$.

Similarly, $Y = \phi(Y/A) \cdot A + R_2$. Since $X < Y$, it follows that $\phi(X/A) \cdot A + R_1 < \phi(Y/A) \cdot A + R_2$ so

1) $[\phi(X/A) - \phi(Y/A)]A < R_2 - R_1$.

Suppose (Y/A) .LT. (X/A). Then $\phi(Y/A) < \phi(X/A)$, and $\phi(X/A) - \phi(Y/A) > 0$. Because these inequalities involve integers, this implies $\phi(X/A) - \phi(Y/A) \geq 1$, and with 1) above gives

2) $0 < A < R_2 - R_1$.

Since X and Y have the same sign, the remainders R_1 and R_2 have the same sign also. The division theorem in C_m^r also shows $|R_1| < |A|$ and $|R_2| < |A|$, so it must be true that $|R_2 - R_1| < A$. This contradicts inequality 2), implying that (Y/A) .LT. (X/A) is false. Hence, (X/A) .LE. (Y/A).

To see that this c-inequality cannot be more strict, consider the following example. Let A be any element of C_m^r, such that 1 .LT. A .LT. Pm. Then take $X = A$ and $Y = A \oplus 1$, so X .LT. Y. The choice of A insures no overflow occurs in deriving Y. The quotients are (X/A) = 1 and (Y/A) = 1, so that (X/A) .EQ. (Y/A).

Theorem 6.14

Let X, Y and A be elements of C_m^r, such that both quotients X/A and Y/A are defined, with A .LT. 0. If X .LT. Y then (X/A) .GE. (Y/A), and this second c-inequality cannot be made more strict.

Proof: This follows in essentially the same way as Theorem 6.13. The counterexample will have Nm .LT. A .LT. -1, with $X = A \ominus 1$ and Y = A, to give X/A .EQ. Y/A .EQ. 1.

In these basic results, the c-inequalities exactly simulate the integer inequalities when overflow is absent. The occurence of overflow causes difficulty. There are many directions which could now be persued with c-inequalities, but will not. Instead, attention will be returned to division in C_m^r, and its interaction with other radix complement operations in more complicated expressions.

CHAPTER 7

A "RATIONAL" ALGEBRA FOR C_m^r

The division operation in the radix complement system is distinct from both of the ring operations addition and multiplication. In particular, it is not the "inverse" of multiplication. Any study of C_m^r, then, should investigate interactions between division and the other two operations. This will be done in two parts. The first considers only division and multiplication.

Division and Multiplication

In a field, division and multiplication are closely related. Division is the "inverse" of multiplication in the sense that it is multiplication by the multiplicative inverse; $\frac{x}{y} = x \cdot y^{-1}$. Division in C_m^r does not have this property; it is a distinct third operation on the ring. Nevertheless, there is some relationship between multiplication and division in C_m^r. Certain properties of field operations are approximated by radix complement operations.

Consider the field properties 1) $\frac{x}{a} \cdot a = x$, 2) $\frac{x \cdot a}{a} = x$ and 3) $\frac{x}{a} = y$ if and only if $x = a \cdot y$, where $a \neq 0$ in each case.

None of these hold in C_m^r. Studying them in C_m^r may yield insight into the relationship between division and multiplication. To simplify notation involving the remainder in division, the function rem will be used; rem(A/B) gives the remainder from the quotient A/B, for A and B in either C_m^r or Z. It is defined when its argument is a defined quotient. The expression rem(A/B) = A ⊖ (A/B)*B gives its value for C_m^r. That is, if A = Q*B ⊕ R and A/B = Q, then rem(A/B) = R.

Theorem 7.1

Let X and A be elements of C_m^r, such that X/A is defined. Then (X/A)*A = X if and only if rem(X/A) = 0.

Proof: The division theorem of Chapter 3 gives X = (X/A)*A ⊕ rem(X/A). The result here follows directly. Note that (X/A)*A does not overflow.

Theorem 7.2

Let X and A be elements of C_m^r. Then (X*A)/A = X if and only if X*A does not overflow.

Proof: If X*A does not overflow, then X*A = X•A. Elements of C_m^r and their signed values are not distinguished. Since (X•A)/A = X in Z, it follows that (X*A)/A = X in C_m^r.

Suppose (X*A)/A = X. The note in the proof above implies that [(X*A)/A]*A does not overflow. Thus X*A does not overflow.

These theorems may be used directly to prove the following.

Theorem 7.3

Let X, Y and A be elements of C_m^r, such that X/A is defined.

1) If rem(X/A) = 0, then X/A = Y implies X = A*Y.

2) If there is no overflow in A*Y, then X = A*Y implies X/A = Y.

Corollary 7.4

Let X, Y and A be elements of C_m^r, such that X/A is defined and exact (rem(X/A) = 0) and Y*A does not overflow. Then X/A = Y if and only if X = A*Y.

The restriction to an exact quotient is quite severe. It is needed because information is lost when the division remainder is discarded. This loss is essentially from the low order significant digits. Overflow represents a loss of information in the high order digits. Overflow loss is generally associated with large magnitude arguments, while remainder loss occurs with all magnitudes.

A result of some utility is included here.

Theorem 7.5

Let X and A be elements of C_m^r, such that X/A is defined and A .GT. 0. Then X ⊖ (X/A)*A = X mod A.

Proof: X ⊖ (X/A)*A = rem(X/A). This is how X mod A is often defined.

Theorem 7.5 also holds for primary division in the integers.

Another property of fields (actually of integral domains) must be restricted in C_m^r. The radix complement system is a ring, but not

an integral domain; divisors of zero do exist ($262144 * 16384 = 0$ in C_{32}^2). This means the cancellation law of multiplication will not generally hold, so $A*X = A*Y$ does not always imply $X = Y$. For example, in C_4^2, $4 * 3 = 4 * 7$, but $3 \neq 7$; in C_{32}^2, $262144 * 16384 = 0 * 16384$, while $262144 \neq 0$. It can be shown that the cancellation property does hold if there is no overflow.

Theorem 7.6

Let X, Y and A be elements of C_m^r, and suppose neither $A*X$ nor $A*Y$ overflow. If $A*X = A*Y$, then $X = Y$.

Proof: Because there is no overflow, $\phi(A*X) = A \cdot X$ and $\phi(X*Y) = A \cdot Y$. Since ϕ is one-to-one, $A \cdot X = A \cdot Y$, so $X = Y$ in both Z and C_m^r.

The Algebra of Fractions

Division in C_m^r approximates the rational field division. For integers x and y, the primary quotient x/y and the rational quotient $\frac{x}{y}$ differ by less than one. The primary quotient has the smaller magnitude because it represents the integer part of the rational quotient. From its definition, division in C_m^r will have the same property.

This suggests another way of investigating the C_m^r operations: evaluate the basic rules of rational arithmetic in C_m^r, particularly those involving division. Quantizing the approximation leads to a non-standard form for expressing rational numbers.

In the work to follow, elements of the integer system Z will be represented using lower case letters, and elements of C_m^r using

upper case letters. An exception to this will occur when no distinction is made between an element A in C_m^r and its signed integer value $\phi(A)$; in this case, A may be either an element of C_m^r or the signed integer value, depending on context. The function $s: Z \to \{1, 0, -1\}$ is the sign function. It is applied to the signed values for elements of C_m^r.

Before considering rules of algebra, a tool of considerable utility will be derived. It is a method of reducing integer equations into the quotient-remainder form $a = q \cdot b + r$ of Chapter 3, where $q = a/b$ and r is the remainder from the division.

Theorem 7.7 (the reduction theorem)

Let a, b, c and d be integers, such that $a = cb + d$ and a/b is defined. Then there exist unique integers f and r such that $a = (c + f)b + r$, where $c + f = a/b$, r has the same sign as a, and $|r| < |b|$. The value of f is given by

$$f = \begin{cases} d/b & \text{if } s(d) = s(a) \text{ or } a = 0 \text{ or } \operatorname{rem}(d/b) = 0, \\ d/b - 1 & \text{if } s(d) \neq s(a) \text{ and } s(a) = s(b), \text{ or} \\ d/b + 1 & \text{if } s(d) \neq s(a) \text{ and } s(a) \neq s(b), \end{cases}$$

where s is the sign function. The value of r is given by $r = d - fb$.

Proof: If the equation $a = cb + d$ can be put into the form $a = (c + f)b + r$, with $s(r) = s(a)$ and $|r| < |b|$, then Theorem 3.3 gives the uniqueness of r and that $c + f = a/b$. Since a/b is unique, f is unique. It must be shown that the transformation can be made.

If either $a = 0$ or $d = 0$, then the result is trivially

true. Suppose $a \neq 0$ and $d \neq 0$. Then the primary division algorithm shows that.

$$d = (d/b)b + r',$$

where $r' = \text{rem}(d/b)$, $s(r') = s(d)$, and $|r'| < |b|$. This means that

$$a = cb + d = cb + (d/b)b + r', \quad \text{or}$$

$$a = (c + d/b)b + r'. \tag{1}$$

If $s(d) = s(a)$, then $s(r') = s(a)$, and the proof is complete with $f = d/b$ and $r = r'$. The same is true if $r' = 0$.

If $r' \neq 0$ and $s(d) \neq s(a)$, then $s(r') \neq s(a)$, so equation (1) above is not quite in the form of the primary division theorem. That form may be achieved by adding $\pm b$ to remainder r', with corresponding adjustment to the other terms. If $s(a) = s(b)$, then $s(b) \neq s(r')$. This means $s(r' + b) = s(a)$ and $|r' + b| < |b|$. Thus, taking $f = d/b - 1$ and $r = r' + b$ completes the proof. If, on the other hand, $s(a) \neq s(b)$, a similar argument will show $f = d/b + 1$ and $r = r' - b$ produce the desired conclusion. The desired value for r is achieved by expanding the right side of $a = (c + f)b + r$ to get $a = cb + (fb + r)$. Because $a = cb + d$, it follows that $r = d - fb$. This proof also shows that the value of r may be given by

$$r = \begin{cases} \text{rem}(d/b) & \text{if } s(d) = s(a) \text{ or } a = 0 \text{ or } \text{rem}(d/b) = 0, \\ \text{rem}(d/b) + b & \text{if } s(d) \neq s(a) \text{ and } s(a) = s(b), \text{ or} \\ \text{rem}(d/b) - b & \text{if } s(d) \neq s(a) \text{ and } s(a) \neq s(b). \end{cases}$$

Theorem 7.8

Under the hypotheses of the preceding theorem, if there exist elements A and B in C_m^r, such that $a = \phi(A)$ and $b = \phi(B)$, and the quotient A/B is defined, then $c + f = \phi(A/B)$ and there is an element R in C_m^r with $r = \phi(R)$.

Proof: The proof of the previous theorem will not carry over to C_m^r because the form $A = Q*B \oplus R$ in C_m^r does not provide a unique value for Q. However, in the integers we have

$$A = cB + d.$$

Reduction in Z now provides

$$A = (c + f)B + r,$$

where $c + f = A/B$, $s(r) = s(A)$ and $|r| < |B|$. It was noted before that $\phi(A/B) = A/B$, so $c + f = \phi(A/B)$. The bound on the magnitude of r shows there is an element R in C_m^r such that $r = \phi(R)$.

Theorem 7.8 gives a reduction theorem in C_m^r paralleling Theorem 7.7 in the integers. For it to work, particularly to provide the uniqueness of f, the intermediate stages must be performed in Z. This will prove useful in many proofs to follow.

The definition and theorems below, taken from the algebra of the rational numbers, will serve as the basis for studying the interaction of division with multiplication and addition. It is assumed that the denominators are not zero.

I. $\frac{A}{B} = \frac{C}{D}$ if and only if $AD = BC$ (Definition)

II. $\frac{A}{B} = \frac{-A}{-B} = -(\frac{-A}{B}) = -(\frac{A}{-B})$

III. $-\frac{A}{B} = -(\frac{-A}{-B}) = \frac{-A}{B} = \frac{A}{-B}$

IV. $\frac{A}{B} = \frac{AC}{BC}$, $C \neq 0$ (Fundamental Principle of Fractions)

V. $\frac{A}{B} + \frac{C}{B} = \frac{A + C}{B}$

VI. $\frac{A}{B} + \frac{C}{D} = \frac{AD + BC}{BD}$

VII. $\frac{A}{B} \cdot \frac{C}{D} = \frac{AC}{BD}$

VIII. $\frac{A}{B} \div \frac{C}{D} = \frac{AD}{BC}$

I. $\frac{A}{B} = \frac{C}{D}$ if and only if $AD = BC$

The definition of equality between rational numbers (I. above) is fundamental to the rational number system, so it might be good to see how closely this property carries over to C_m^r. Substantial modification can be expected; the loss of the remainders from the A/B and C/D suggest this, even before considering overflow in A*D and B*C.

Theorem 7.9

Let A, B, C and D be elements of C_m^r, such that both the quotients A/B and C/D are defined. Let R_1 and R_2 be the respective remainders, and suppose that none of the products A*D, B*C, R_1*D or R_2*B overflow. Then

A*D = B*C if and only if A/B = C/D and R_1*D = R_2*B.

Before proving this theorem, it may help to have a lemma concerning the rational numbers. This will also provide further insight into the nature of the primary division operation.

Lemma 7.10

Let a and b be integers, with $b \neq 0$, and let r be the remainder from a/b. Then $\frac{a}{b} = (a/b) + \frac{r}{b}$.

Proof: Note that a/b is an integer, and by the primary division theorem $a = (a/b)b + r$. This is an integer equation, but taken as an equation in the rationals, both sides may be divided by b to give $\frac{a}{b} = (a/b) + \frac{r}{b}$.

The form $(a/b) + \frac{r}{b}$ is just the "mixed" form of the rational number $\frac{a}{b}$, and the primary quotient of two integers is, as noted before, the integer part of that rational quotient. With this in mind, it follows that for integers a, b, c and d ($b \neq 0$ and $d \neq 0$), we have $\frac{a}{b} = \frac{c}{d}$ if and only if $a/b = c/d$ and $\frac{r_1}{b} = \frac{r_2}{d}$, where r_1 and r_2 are the remainders from a/b and c/d, respectively. Furthermore, $\frac{r_1}{b} = \frac{r_2}{d}$ if and only if $r_1 d = r_2 b$, so that

(1) $\frac{a}{b} = \frac{c}{d}$ if and only if $a/b = c/d$ and $r_1 d = r_2 b$.

Proof of Theorem 7.9: From the no-overflow conditions, $A*D = B*C$ is equivalent to $A \cdot D = B \cdot C$. This in turn is equivalent to $\frac{A}{B} = \frac{C}{D}$, which by (1) above, is the same as $A/B = C/D$ and $R_1 D = R_2 B$. Again, by the no-overflow conditions, and because $\phi(A/B) = A/B$ and $\phi(C/D) = C/D$, this last is equivalent to $A/B = C/D$ and $R_1 * D = R_2 * B$.

The chain of equivalences gives

$$A*D = B*C \text{ if and only if } A/B = C/D \text{ and } R_1*D = R_2*B.$$

The overflow conditions required in Theorem 7.9 seem fairly severe. They can be weakened considerably if the theorem is not stated as an equivalence.

Theorem 7.11

Let A, B, C and D be elements of C_m^r, such that both A/B and C/D are defined. Let R_1 and R_2 be the respective remainders.

1) If the modulo reduction factors of A*D and B*C are the same, then A*D = B*C implies A/B = C/D and $R_1*D = R_2*B$.

2) A/B = C/D and $R_1*D = R_2*B$ imply A*D = B*C.

Proof: 1) Let k_1 and k_2 be the modulo reduction factors for A*D and B*C, respectively, so that $A*D = A\cdot D + k_1 r^m$ and $B*C = B\cdot C + k_2 r^m$. Then A*D = B*C implies that $A\cdot D + k_1 r^m = B\cdot C + k_2 r^m$. By hypothesis, $k_1 = k_2$, so $A\cdot D = B\cdot C$. From the proof of Theorem 7.9, this implies that A/B = C/D and $R_1 D = R_2 B$. Because of the ring homomorphism into C_m^r, this will make $R_1*D = R_2*B$, so A/B = C/D and $R_1*D = R_2*B$.

2) The division theorem in C_m^r gives $A = (A/B)*B \oplus R_1$ and $C = (C/D)*D \oplus R_2$. Then $A*D = (A/B)*B*D \oplus R_1*D$ and $B*C = (C/D)*B*D \oplus R_2*B$. Because A/B = C/D, it is true that (A/B)*B*D = (C/D)*B*D. The hypothesis provides $R_1*D = R_2*B$, so A*D = B*C.

The need in part 1) of Theorem 7.11, to have both modulo reduc-

tion factors the same is shown in this example.

Example 7.12

Consider the four bit 2's complement system, $C_4^2 = \{-8, -7, \ldots, 6, 7\}$. Let $A = 5$, $B = 4$, $C = 1$ and $D = 4$. Then $A/B = 1$ with remainder $R_1 = 1$, and $C/D = 0$ with remainder $R_2 = 1$. Note that $A*D = 5 * 4 = 4$, with modulo reduction factor $k_1 = -1$, and $B*C = 4 * 1 = 4$, with $k_2 = 0$. The remainders give $R_1*D = 1 * 4 = R_2*B$. Thus $A*D = B*C$ and $R_1*D = R_2*B$, while $A/B \neq C/D$. Note that $k_1 \neq k_2$.

Most users assume the radix complement system is (or at least simulates) a subset of the integers, and that $C_m^r \times C_m^r$ simulates a subset of $Z \times Z$. It might then be possible for $C_m^r \times C_m^r$ to model a subset of the rational numbers. The work above does not deny the possibility, when overflow is absent. Theorem 7.9 also suggests that a subset of $C_m^r \times C_m^r \times C_m^r$ might be used for the same purpose. One component of the ordered triple corresponding to $\frac{A}{B}$ would represent the integer part A/B, another component the remainder, and the third, the denominator B. All three would be needed to test equality of triples, as indicated in Theorem 7.9. Such a set of ordered triples would simulate the rationals in mixed form. This will be considered shortly.

II. $\frac{A}{B} = \frac{-A}{-B} = -(\frac{-A}{B}) = -(\frac{A}{-B})$

III. $-\frac{A}{B} = -(\frac{-A}{-B}) = \frac{-A}{B} = \frac{A}{-B}$

These properties translate almost directly into the radix complement system, providing the divisions are defined. Differences arise

when C_m^r is unbalanced, because then $Nm = \ominus Nm$.

Theorem 7.13

Let A and B be elements of C_m^r, such that A/B is defined.

1) If C_m^r is balanced, or if C_m^r is unbalanced and $A \neq Nm$, then

$$A/B = \ominus A/\ominus B = \ominus(\ominus A/B) = \ominus(A/\ominus B) \qquad \text{and}$$

$$\ominus(A/B) = \ominus(\ominus A/\ominus B) = \ominus A/B = A/\ominus B.$$

2) If C_m^r is unbalanced, $A = Nm$, and $B \neq Nm$, then

$$A/B = \ominus(\ominus A/\ominus B) = \ominus A/B = \ominus(A/\ominus B), \qquad \text{and}$$

$$\ominus(A/B) = \ominus A/\ominus B = \ominus(\ominus A/B) = A/\ominus B.$$

3) If C_m^r is unbalanced and $A = B = Nm$, then

$$A/B = \ominus A/\ominus B = \ominus A/B = A/\ominus B, \qquad \text{and}$$

$$\ominus(A/B) = \ominus(\ominus A/\ominus B) = \ominus(\ominus A/B) = \ominus(A/\ominus B).$$

To avoid ambiguity, the unary minus sign $\ominus$ is taken to have higher precedence than any other arithmetic operator; it will be applied before any other operator. This reduces the number of parentheses required in an expression.

Proof: 1) If the system is unbalanced and $B = Nm$, then all the quotients are zero, yielding both sets of equalities. If the system is balanced, or if it is unbalanced and $B \neq Nm$, then both $\phi(\ominus A) = -A$ and $\phi(\ominus B) = -B$. Since $\ominus A/B \neq Nm$ and $A/\ominus B \neq Nm$, it

follows that $\phi[\theta(\theta A/B)] = -\phi(\theta A/B)$ and $\phi[\theta(A/\theta B)] = -\phi(A/\theta B)$. Consider the four integer quotients A/B, $-A/-B$, $-(-A/B)$ and $-(A/-B)$. To see that these are equal note that the primary quotient x/y takes as magnitude the integer part of $\frac{|x|}{|y|}$. This means all four quotients have the same magnitude. If the magnitude of x/y is not zero, its sign is taken to be the algebraic sign of the rational number $\frac{x}{y}$. Thus $A/B = -A/-B = -(-A/B) = -(A/-B)$ in Z, and $A/B = \theta A/\theta B = \theta(\theta A/B) = \theta(A/\theta B)$ in C_m^r. The other set of equalities is derived by negating each term.

2) In an unbalanced system, with $A = Nm$, we have $A = \theta A$. Thus $\theta A/\theta B = A/\theta B$ and $\theta A/B = A/B$. Since $B \neq -1$ (else A/B is not defined), the sign and magnitude argument of part 1) applies to give $A/B = \theta(\theta A/\theta B) = \theta A/B = \theta(A/\theta B)$. Negating each term again provides the second set of equalities.

3) If $A = B = Nm$ in an unbalanced system, then both $A = \theta A$ and $B = \theta B$. This means $A/B = \theta A/\theta B = \theta A/B = A/\theta B$, where each quotient is equal to one. The second set of equalities is obtained as before.

$$\text{IV.}\quad \frac{A}{B} = \frac{AC}{BC},\ C \neq 0$$

This property is known as the Fundamental Principle of Fractions for a field. It is only approximately true in the radix complement system if there is no overflow; in this case, the two expressions will differ by no more than one. Overflow changes this considerably.

The difference between A/B and $(A*C)/(B*C)$ may be characterized in more than one way. It may be taken as an element of C_m^r, given by subtracting the expressions, or it may be given by subtracting

the signed values of the expressions as $\phi(A/B) - \phi\{(A*C)/(B*C)\}$. The value in C_m^r is equivalent modulo r^m to the integer value, so this latter will be developed first.

Theorem 7.14

Let A, B and C be elements of C_m^r, such that both A/B and (A*C)/(B*C) are defined. Let k_1 and k_2 be the modulo reduction factors for A*C and B*C, respectively, and let R be the remainder from (A*C)/(B*C). To simplify notation, let d be the integer $d = (R + k_2 r^m \phi\{(A*C)/(B*C)\} - k_1 r^m)/C$. Then

$$\phi(A/B) = \phi\{(A*C)/(B*C)\} + f,$$

where

$$f = \begin{cases} d/B & \text{if } s(d) = s(A), \text{ or } A = 0, \text{ or } \text{rem}(d/B) = 0 \\ d/B - 1 & \text{if } s(d) \neq s(A) \text{ and } s(A) = s(B), \text{ or} \\ d/B + 1 & \text{if } s(d) \neq s(A) \text{ and } s(A) \neq s(B). \end{cases}$$

Proof: Let $Q = (A*C)/(B*C)$. Then the primary division theorem implies

$$\phi(A*C) = Q \cdot \phi(B*C) + R.$$

Introducing the modulo reduction factors k_1 and k_2 gives

$$A \cdot C + k_1 r^m = Q(B \cdot C + k_2 r^m) + R,$$

which simplifies to

$$A \cdot C = Q \cdot B \cdot C + (R + k_2 r^m Q - k_1 r^m).$$

Since this is an equation in the integers, and two of the terms are evenly divisible by C, the third term must also be divisible by C.

Thus

$$A = Q \cdot B + (R + k_2 r^m Q - k_1 r^m)/C.$$

Note that the right-hand term is the integer d used in the statement of the theorem. The reduction theorem 7.7 now implies there is an integer f, given by the expression above, such that $(Q + f) = A/B$. This is the same as

$$\phi(A/B) = \phi\{(A*C)/(B*C)\} + f,$$

and the proof is complete.

The next lemma is for the special case when there is no overflow.

Lemma 7.15

Let x, y and c be integers. If $|x| < |yc|$, then $|x/c| < |y|$.

Proof: Note the hypothesis implies $c \neq 0$ and $y \neq 0$. From the primary division theorem, $x = (x/c)c + r$, where $s(r) = s(x)$ and $|r| < |c|$. If $|x| < |yc|$, then

$$|(x/c)c + r| < |yc|.$$

Consider this integer inequality as a rational inequality. Dividing both sides by $|c|$ gives

$$(1) \qquad |(x/c) + \frac{r}{c}| < |y|.$$

If $r = 0$, the conclusion $|x/c| < |y|$ follows immediately from (1). It also follows if $x/c = 0$, because $y \neq 0$. Suppose neither x/c nor r are zero. Since x and r have the same sign, x/c and $\frac{r}{c}$ also have like signs. Thus

$$|x/c| < |(x/c) + \frac{r}{c}|,$$

and combining this with (1) gives $|x/c| < |y|$.

Corollary 7.16

Under the hypotheses of Theorem 7.14, if neither $A*C$ nor $B*C$ overflow, so that $k_1 = k_2 = 0$, then

$$\phi(A/B) = \phi\{(A*C)/(B*C)\} + f,$$

where

$$f = \begin{cases} 0 & \text{if } s(R/C) = s(A), \text{ or } A = 0, \text{ or } R/C = 0, \\ 1 & \text{if } s(R/C) \neq s(A) \text{ and } s(A) = s(B), \text{ or} \\ -1 & \text{if } s(R/C) \neq s(A) \text{ and } s(A) \neq s(B). \end{cases}$$

Proof: Since $k_1 = k_2 = 0$, Theorem 7.14 applies with $d = R/C$. Because R is the remainder from $(A*C)/(B*C)$, we have $|R| < |\phi(B*C)|$. The no-overflow condition means $\phi(B*C) = B \cdot C$, so $|R| < |B \cdot C|$. By Lemma 7.15, this implies $|R/C| < |B|$, which gives $d/B = (R/C)/B = 0$ and $\text{rem}(d/B) = R/C$. Using this in Theorem 7.14 yields the desired form for f.

It might be noted that Corollary 7.16 also applies to the integers; integer products never overflow.

Corollary 7.17

Under the hypotheses of Theorem 7.14, and with the integer f given there,

$$(A/B) \ominus \{(A*C)/(B*C)\} = \psi(f).$$

Proof: ψ is the ring homomorphism from Z to C_m^r with $\psi \circ \phi$ the identity map.

$$\text{V.}\quad \frac{A}{B} + \frac{C}{B} = \frac{A + C}{B}$$

$$\text{VI.}\quad \frac{A}{B} + \frac{C}{D} = \frac{AD + BC}{BD}$$

These are the addition rules for fractions, and it will be seen that they are also approximately true in C_m^r when overflow is absent.

Theorem 7.18

Let A, B and C be elements of C_m^r, such that A/B, C/B and $(A \oplus C)/B$ are defined. Let R_1 and R_2 be the remainders from A/B and C/B, respectively, and let k be the modulo reduction factor for $A \oplus C$. If $d = R_1 + R_2 + kr^m$, then

$$\phi\{(A \oplus C)/B\} = \phi(A/B) + \phi(C/B) + f,$$

where

$$f = \begin{cases} d/B & \text{if } s(d) = s(A \oplus C) \text{ or } A \oplus C = 0 \text{ or } \mathrm{rem}(d/B) = 0, \\ d/B - 1 & \text{if } s(d) \neq s(A \oplus C) \text{ and } s(A \oplus C) = s(B), \text{ or} \\ d/B + 1 & \text{if } s(d) \neq s(A \oplus C) \text{ and } s(A \oplus C) \neq s(B). \end{cases}$$

Proof: The primary division theorem allows us to write $A = \phi(A/B)B + R_1$ and $C = \phi(C/B)B + R_2$. The definition of ϕ yields

$$\phi(A \oplus C) = A + C + kr^m.$$

Substituting,

$$\phi(A \oplus C) = \phi(A/B)B + R_1 + \phi(C/B)B + R_2 + kr^m,$$

which simplifies to

$$\phi(A \oplus C) = \{\phi(A/B) + \phi(C/B)\}B + (R_1 + R_2 + kr^m).$$

The right-hand term is d, and $\phi(A \oplus C)/B = \phi\{(A \oplus C)/B\}$, so Theorem 7.7 provides the conclusion.

Corollary 7.19

If no overflow occurs in $A \oplus C$ and the hypotheses of Theorem 7.18 are satisfied, then

$$\phi\{(A \oplus C)/B\} = \phi(A/B) + \phi(C/B) + f,$$

where f takes on one of the values $+1$, 0 or -1.

Proof: This comes from Theorem 7.18, with the modulo reduction factor $k = 0$, and $d = R_1 + R_2$. Note that $|d/B| \leq 1$. If $d/B \neq 0$, then the magnitude restrictions on R_1 and R_2 imply $s(R_1) = s(R_2) \neq 0$. This means $s(A) = s(C)$, so that $s(d) = s(R_1 + R_2) = s(A \oplus C)$. The contrapositive of this argument is that $s(d) \neq s(A \oplus C)$ implies $d/B = 0$. Hence, the values of f given by the expression in Theorem 7.18 must be $+1$, 0 or -1.

Corollary 7.19 shows again that, in the absence of overflow, field properties may hold approximately in the radix complement system.

Corollary 7.20

$$(A \oplus C)/B = (A/B) \oplus (C/B) \oplus \psi(f)$$

Theorem 7.18 may be proved in a completely different way. The argument does not require Theorem 7.7, but is algebraic and rather tedious, involving the signs and relative magnitudes of various quantities.

Theorem 7.21

Let A, B, C and D be elements of C_m^r, such that

A/B, C/D, and $[(A*D) \oplus (B*C)]/(B*D)$ are defined. Let k_1 be the modulo reduction factor for the expression $(A*D) \oplus (B*C)$, and k_2 be the m.r.f. for $B*C$. If R_1 and R_2 are the remainders for A/B and C/D, respectively, then

$$\phi\{[(A*D) \oplus (B*C)]/(B*D)\} = \phi(A/B) + \phi(C/D) + f,$$

where

$$f = \begin{cases} d/\phi(B*D) & \text{if } s[(A*D) \oplus (B*C)] = s(d) \text{ or } (A*D) \oplus (B*C) = 0 \\ & \quad \text{or } \operatorname{rem}(d/\phi(B*D)) = 0, \\ d/\phi(B*D) - 1 & \text{if } s[(A*D) \oplus (B*C)] \neq s(d) \text{ and } s[(A*D) \oplus (B*C)] = s(B*D), \\ d/\phi(B*D) + 1 & \text{if } s[(A*D) \oplus (B*C)] \neq s(d) \text{ and } s[(A*D) \oplus (B*C)] \neq s(B*D), \end{cases}$$

with $d = R_1D + R_2B + \{k_1 - k_2[\phi(A/B) + \phi(C/D)]\}r^m$.

Proof: This will follow the pattern of Theorem 7.18. For simplicity, let $q_1 = \phi(A/B)$ and $q_2 = \phi(C/D)$. The primary division theorem gives $A = q_1B + R_1$ and $C = q_2D + R_2$. With

$$\phi[(A*D) \oplus (B*C)] = A \cdot D + B \cdot C + k_1 r^m,$$

substitution gives

$$\phi[(A*D) \oplus (B*C)] = (q_1B + R_1)D + (q_2D + R_2)B + k_1 r^m.$$

Simplifying this,

$$\phi[(A*D) \oplus (B*C)] = (q_1 + q_2)B \cdot D + (R_1D + R_2B + k_1 r^m).$$

Since $\phi(B*D) = B \cdot D + k_2 r^m$, we have $B \cdot D = \phi(B*D) - k_2 r^m$, so

$$\phi[(A*D) \oplus (B*C)] = (q_1 + q_2)[\phi(B*D) - k_2 r^m] + (R_1D + R_2B + k_1 r^m).$$

Another simplification gives

$$\phi[(A*D) \oplus (B*C)] = (q_1 + q_2) \cdot \phi(B*D) + [R_1D + R_2B + k_1 r^m - k_2(q_1 + q_2) r^m].$$

The right-most term is the integer d above. Taking note of the definitions of q_1 and q_2, Theorem 7.7 again completes the proof.

Corollary 7.22

If there is no overflow in Theorem 7.21, then

$$\phi\{[(A*D) \oplus (B*C)]/(B*D)\} = \phi(A/B) + \phi(C/D) + f,$$

where f takes a value +1, 0 or -1.

Proof: The condition of "no overflow" may be interpreted two ways. It may mean the modulo reduction factors k_1 and k_2 in Theorem 7.21 are both zero, or that none of the radix complement operations in the equation overflow. The first case is more general. If none of the operations overflow, then $k_1 = k_2 = 0$; the converse does not hold. The weaker interpretation will be taken.

Assume $k_1 = k_2 = 0$. Then $\phi\{(A*D) \oplus (B*C)\} = A \cdot D + B \cdot C$ and $\phi(B*D) = B \cdot D$. Theorem 7.21 gives $d = R_1 D + R_2 B$. Note that $|(R_1 \cdot D + R_2 B)/(B \cdot D)| \leq 1$, because $|R_1| < |B|$ and $|R_2| < |D|$.

Suppose $(R_1 \cdot D + R_2 \cdot B)/(B \cdot D) \neq 0$. Since R_1 and R_2 are remainders, $s(R_1) = s(A)$ and $s(R_2) = s(C)$. This means $s(R_1 \cdot D) = s(A \cdot D)$ and $s(R_2 \cdot B) = s(C \cdot B)$, so $s(A \cdot D) = s(B \cdot C)$. Thus $s(d) = s(R_1 \cdot D + R_2 \cdot B) = s(A \cdot D + B \cdot C) = s\{(A*D) \oplus (B*C)\}$. The contrapositive of this says that $s(d) \neq s\{(A*D) \oplus (B*C)\}$ implies $d/\phi(B*D) = 0$. Therefore, considering the bounds for $d/\phi(B*D)$, the value of f will be +1, 0 or -1, given by the expression in Theorem 7.21.

Corollary 7.23

$$\{[(A*D) \oplus (B*C)]/(B*D)\} \ominus [(A/B) \oplus (C/D)] = \psi(f).$$

In the no-overflow case, $\psi(f) = +1, 0$ or -1. Moreover, this value does not involve overflow in the subtraction.

VII. $\frac{A}{B} * \frac{C}{D} = \frac{AC}{BD}$

This property, and property VIII, will not translate to the radix complement system with the facility of those already considered. Even in the no-overflow case, the values of the expressions (A/B)*(C/D) and (A*C)/(B*D) may differ by quite significant amounts. For example, (1/2)*(Pm/1) = 0 while (1*Pm)/(2*1) = Pm/2. On the IBM/370, $Pm = P_{32}^2 = 2147483647$, and these two expressions become (1/2)*(2147483647/1) = 0 and (1*2147483647)/(2*1) = 1073741823. The problem occurs because division remainder is lost. When two quotients are multiplied, this loss, in effect, is multiplied also. In previous theorems, the loss was additive. Overflow, as usual, further complicates the picture.

Theorem 7.24

Let A, B, C and D be elements of C_m^r, such that A/B, C/D and (A*C)/(B*D) are defined. Let k_1 and k_2 be the modulo reduction factors for A*C and B*D, respectively, and R_1 and R_2 be the remainders from A/B and C/D, respectively. Define the integer d by

$$d = \phi(A/B)B \cdot R_2 + \phi(C/D)D \cdot R_1 + R_1 \cdot R_2 + k_1 r^m - k_2 \phi(A/B)\phi(C/D) r^m.$$

Then $\phi[(A*C)/(B*D)] = \phi(A/B)\phi(C/D) + f$,

where

$$f = \begin{cases} d/\phi(B*D) & \text{if } s(d) = s(A*C), \text{ or } A*C = 0, \text{ or } \mathrm{rem}(d/\phi(B*D)) = 0, \\ d/\phi(B*D) - 1 & \text{if } s(d) \neq s(A*C), \text{ and } s(A*C) = s(B*D), \text{ or} \\ d/\phi(B*D) + 1 & \text{if } s(d) \neq s(A*C), \text{ and } s(A*C) \neq s(B*C). \end{cases}$$

Proof: This will follow the established pattern. We have $A = \phi(A/B)B + R_1$ and $C = \phi(C/D)D + R_2$. Furthermore, $\phi(A*C) = A \cdot C + k_1 r^m$ and $\phi(B*D) = B \cdot D + k_2 r^m$, the latter giving $B \cdot D = \phi(B*D) - k_2 r^m$. Elementary algebraic manipulation then yields

$$\phi(A*C) = [\phi(A/B)B + R_1][\phi(C/D)D + R_2] + k_1 r^m,$$

$$\phi(A*C) = [\phi(A/B)\phi(C/D)]B \cdot D + [\phi(A/B)B \cdot R_2 + \phi(C/D)D \cdot R_1 + R_1 R_2 + k_1 r^m],$$

$$\phi(A*C) = [\phi(A/B)\phi(C/D)][\phi(B*D) - k_2 r^m] + [\phi(A/B)B \cdot R_2 + \phi(C/D)D \cdot R_1 + R_1 R_2 + k_1 r^m],$$

and

$$\phi(A*C) = [\phi(A/B)\phi(C/D)]\phi(B*D) + d,$$

where d is the given integer expression. Theorem 7.7 provides the conclusion.

The form of the "pseudo-remainder" d is somewhat more cumbersome than before, particularly that part involving the modulo reduction factors. It can be simplified by noting that $\phi(A/B)B = A - R_1$ and $\phi(C/D)D = C - R_2$. Substituting these gives

$$d = A \cdot R_2 + C \cdot R_1 - R_1 R_2 + k_1 r^m - k_2 \phi(A/B)\phi(C/D) r^m.$$

Corollary 7.25

If neither $A*C$ nor $B*D$ overflow in Theorem 7.24, then

$$\phi[(A*C)/(B*D)] = \phi(A/B)\phi(C/D) + f,$$

where

$$f = \begin{cases} (A\cdot R_2 + C\cdot R_1 - R_1R_2)/(B\cdot D) & \text{if } s(d) = s(A*C), \text{ or } A*C = 0 \\ & \quad \text{or } \operatorname{rem}(d/(B\cdot D)) = 0, \\ (A\cdot R_2 + C\cdot R_1 - R_1R_2)/(B\cdot D) - 1 & \text{if } s(d) \neq s(A*C) \text{ and} \\ & \quad s(A*C) = s(B*D), \text{ or} \\ (A\cdot R_2 + C\cdot R_1 - R_1R_2)/(B\cdot D) + 1 & \text{if } s(d) \neq s(A*C) \text{ and} \\ & \quad s(A*C) \neq s(B*D). \end{cases}$$

This expression for f shows how the loss of remainders is multiplied. Even in the no-overflow case, the value of f may be large.

To investigate a maximum value for f in Corollary 7.25, consider the original form of the "pseudo-remainder" d in Theorem 7.24, $d = \phi(A/B)B\cdot R_2 + \phi(C/D)D\cdot R_1 + R_1\cdot R_2$. ($k_1 = k_2 = 0$ in the no-overflow case.) The rational number

$$(1) \qquad \frac{d}{B\cdot D} = \frac{\phi(A/B)R_2}{D} + \frac{\phi(C/D)R_1}{B} + \frac{R_1\cdot R_2}{B\cdot D}$$

is, in magnitude, an upper bound for the magnitude of the integer $d/(B\cdot D)$; $\frac{d}{B\cdot D}$ and $d/(B\cdot D)$ differ by less than one. Thus, the maximum value of $\frac{d}{B\cdot D}$ will give the maximum for $d/(B\cdot D)$ and, in turn, the maximum for f.

If $\frac{d}{B\cdot D}$ is to be maximal, then all three terms on the right of equation (1) must have the same sign. For simplicity, consider only the case where A, B, C and D are positive, so each term in (1) will be non-negative. The remainders R_1 and R_2 will then both be non-negative, and the terms in (1) will be maximal when the remainders have their greatest values. This will happen when $R_1 = B - 1$ and $R_2 = D - 1$. Substituting these values into expression (1) and simpli-

fying gives

$$(2)\quad \frac{d}{B\cdot D} = \phi(A/B)\left(1 - \frac{1}{D}\right) + \phi(C/D)\left(1 - \frac{1}{B}\right) + \left(1 - \frac{1}{B}\right)\left(1 - \frac{1}{D}\right).$$

Since all the terms are non-negative, equation (2) shows that

$$(3)\quad \frac{d}{B\cdot D} < \phi(A/B) + \phi(C/D) + 1.$$

Rewriting (2) another way gives

$$\frac{d}{B\cdot D} = \{\phi(A/B) + \phi(C/D) + 1\}\left(1 - \frac{1}{D}\right)\left(1 - \frac{1}{B}\right) + \frac{\phi(A/B)}{B}\left(1 - \frac{1}{D}\right) + \frac{\phi(C/D)}{D}\left(1 - \frac{1}{B}\right).$$

Again, the terms are non-negative, so omitting two of them gives a lower bound,

$$(4)\quad \frac{d}{B\cdot D} \geq \{\phi(A/B) + \phi(C/D) + 1\}\left(1 - \frac{1}{B}\right)\left(1 - \frac{1}{D}\right).$$

Equality may occur in (4) when $B = D = 1$, because then $d = 0$. Combining inequalities (3) and (4) shows

$$\{\phi(A/B) + \phi(C/D) + 1\}\left(1 - \frac{1}{B}\right)\left(1 - \frac{1}{D}\right) \leq \frac{d}{B\cdot D} \leq \phi(A/B) + \phi(C/D) + 1.$$

The following result has been established.

Theorem 7.26

Let A, B, C and D be non-negative elements of C_m^r satisfying the hypotheses of Corollary 7.25. Then the value of f given there is strictly bounded above by $\phi(A/B) + \phi(C/D) + 1$. The maximal value of f, achieved when R_1 and R_2 are maximal, is bounded below by

$$\{\phi(A/B) + \phi(C/D) + 1\}\left(1 - \frac{1}{B}\right)\left(1 - \frac{1}{D}\right).$$

Note that the maximal value of f in Corollary 7.25 will approach $\phi(A/B) + \phi(C/D) + 1$ when B and D are large. At the same

time, both A/B and C/D will be small for large values of B and D.

Corollary 7.27

Let A, B, C and D be elements of C_m^r satisfying the hypotheses of Theorem 7.24. Then

$$(A*C)/(B*D) = (A/B)*(C/D) \oplus \psi(f),$$

where f is the integer expression given in the theorem.

VIII. $\frac{A}{B} \div \frac{C}{D} = \frac{AD}{BC}$

For this property, the standard technique will not be used. It would lead to a very difficult derivation, with the resulting formulation so involved as to impede evaluation.

Let d be the integer such that

(1) $\phi(A/B) = \phi[(A*D)/(B*C)]\cdot\phi(C/D) + d.$

When A/B, C/D and (A/B)/(C/D) are defined, none of C, D or C/D may be zero. This means D/C will be defined, and Theorem 7.24 says that

$$\phi[(A*D)/(B*C)] = \phi(A/B)\cdot\phi(D/C) + g,$$

where g is the integer expression given in the theorem. Substituting this into equation (1) gives

$$\phi(A/B) = \phi(A/B)\cdot\phi(D/C)\cdot\phi(C/D) + g\cdot\phi(C/D) + d.$$

Since $\phi(D/C)\phi(C/D) = \begin{cases} 0 & \text{if } |C| \neq |D|, \text{ or} \\ 1 & \text{if } |C| = |D|, \end{cases}$

it follows that

$$\phi(A/B) = \begin{cases} g\cdot\phi(C/D) + d & \text{if } |C| \neq |D|, \\ g\cdot\phi(C/D) + d + \phi(A/B) & \text{if } |C| = |D|. \end{cases}$$

Solving this for d yields

$$d = \begin{cases} \phi(A/B) - g\cdot\phi(C/D) & \text{if } |C| \neq |D|, \\ -g\cdot\phi(C/D) & \text{if } |C| = |D|. \end{cases}$$

Note that $|C| \geq |D|$, because otherwise, $C/D = 0$. With this expression for d, Theorem 7.7 is applied to equation (1), proving

Theorem 7.28

Let A, B, C and D be elements of C_m^r, such that A/B, C/D and $(A/B)/(C/D)$ are defined. Let g be the integer expression defined in Theorem 7.24 such that $\phi[(A*D)/(B*C)] = \phi(A/B)\phi(D/C) + g$. Then

$$\phi[(A/B)/(C/D)] = \phi[(A*D)/(B*C)] + f,$$

where

$$f = \begin{cases} d/\phi(C/D) & \text{if } s(d) = s(A/B), \text{ or } A/B = 0 \\ & \quad \text{or } \operatorname{rem}(d/\phi(C/D)) = 0, \\ d/\phi(C/D) - 1 & \text{if } s(d) \neq s(A/B) \text{ and } s(A/B) = s(C/D), \text{ or} \\ d/\phi(C/D) + 1 & \text{if } s(d) \neq s(A/B) \text{ and } s(A/B) \neq s(C/D), \end{cases}$$

with

$$d = \begin{cases} \phi(A/B) - g\cdot\phi(C/D) & \text{if } |C| \neq |D|, \\ -g\cdot\phi(C/D) & \text{if } |C| = |D|. \end{cases}$$

The expression for d involves the integer expression g quite directly, and indicates that the values of f and g should be linked rather closely. In the case where $|C| = |D|$, it follows that $f = -g + t$, where t may be -1, 0, or $+1$. If $|C| \neq |D|$, a bit more work is involved, but Corollaries 7.19 and 7.20 of Theorem 7.18 may be used to show $f = \phi(A/B)/\phi(C/D) - g + t$, where t may again be $+1$, 0 or -1. Maximal values of g occur when $\phi(A/B)$ and $\phi(C/D)$ both have large magnitude and the same signs. In this situation, $\phi(A/B)/\phi(C/D)$ will have relatively small magnitude, indicating f may attain magnitudes approximately equal to those attained by g.

When overflow does not occur in either $A*D$ or $B*C$, the statement of Theorem 7.28 will suffice, except that the integer expression g must be from Corollary 7.25 (or Theorem 7.24 with all modulo reduction factors zero).

Corollary 7.29

Let A, B, C and D be elements of C_m^r satisfying the hypotheses of Theorem 7.28. Then

$$(A/B)/(C/D) = (A*D)/(B*C) \oplus \psi(f),$$

where f is the integer expression given in that theorem.

The properties shown here all involve division and a second radix complement operation. If the second operation is addition or subtraction, and if overflow does not occur, then the radix complement property resembles the corresponding field property, except for a small additive constant. If overflow occurs, or if the second operation is multiplication or division, the radix complement and field properties

differ greatly.

It is important to note that some results in this chapter are true in the system of integers. When there is no overflow, the four arithmetic operations in C_m^r are equivalent to the corresponding operations in Z. Each corollary for the no-overflow case applies directly to the integers. This will be exploited to develop a system for mixed form rational numbers. Another application will be seen first, however.

The radix complement system is often used to simulate the integers without regard to the modular nature of C_m^r. Overflow is considered to be error. The next chapter will quantize this "error."

CHAPTER 8

ERROR IN C_m^r CALCULATIONS

When the radix complement system is used in place of the integers, overflow often is considered an error by users. If an expression over C_m^r involves only the operations of addition, subtraction, and multiplication, any such error can be evaluated using the modulo reduction factors in Definition 2.10. The error will be some multiple of r^m. The situation is more complicated if division is involved.

The problem may be stated in the following way. Suppose E_c is an arithmetic expression over C_m^r, involving operations of addition, subtraction, multiplication and/or division. The expression E_z will be called the corresponding integer expression to E_c if it is obtained in this way; each element of C_m^r appearing in E_c is replaced by its signed value, and each operation $\oplus$, $\ominus$, * or / is replaced by the corresponding integer operation +, -, • or /. If no distinction is made between an element A of C_m^r and its signed value $\phi(A)$, the first part of this correspondence does nothing.

Definition 8.1

Let X be the result of evaluating E_c in C_m^r, and let x be the value of E_z. The error in E_c is given by the integer $x - \phi(X)$.

As an example, consider the expression $A \oplus B$ in C_m^r. Its corresponding integer expression is $\phi(A) + \phi(B)$. The error in $A \oplus B$ is then given by $\phi(A) + \phi(B) - \phi(A \oplus B)$. Lemmas 6.1, 6.2 and 6.3 show $\phi(A \oplus B) = \phi(A) + \phi(B) + kr^m$, where k takes a value +1, 0 or -1, so the error is $-kr^m$. When $A \oplus B$ has positive overflow, $k = -1$ and the error is r^m; negative overflow gives an error of $-r^m$.

The next theorems describe the error when two expressions, each with error, are combined using one of the ring operations $\oplus$, $\ominus$ or $*$. In each case, E_1 and E_2 are expressions over C_m^r having errors e_1 and e_2, respectively. F_1 and F_2 are the corresponding integer expressions. Note that $e_1 = F_1 - \phi(E_1)$ and $e_2 = F_2 - \phi(E_2)$, and no distinction is made between an expression and its value.

Theorem 8.2

The error in $E_1 \oplus E_2$ is $e_1 + e_2 - kr^m$, where k is the modulo reduction factor for $E_1 \oplus E_2$.

Proof: From Definition 8.1, the error in $E_1 \oplus E_2$ is given by $F_1 + F_2 - \phi(E_1 \oplus E_2)$. Previous work has shown

$$\phi(E_1 \oplus E_2) = \phi(E_1) + \phi(E_2) + kr^m,$$

where k is the modulo reduction factor for $E_1 \oplus E_2$. Combining these and using the definitions of e_1 and e_2 gives the error in $E_1 \oplus E_2$ as $e_1 + e_2 - kr^m$.

Theorem 8.3

The error in $E_1 \ominus E_2$ is $e_1 - e_2 - kr^m$, where k is the modulo reduction factor for $E_1 \ominus E_2$.

Proof: This is essentially the same as in Theorem 8.2.

Theorem 8.4

The error in $E_1 * E_2$ is $e_1F_2 + e_2F_1 - e_1e_2 - kr^m$, where k is the modulo reduction factor for $E_1 * E_2$.

Proof: The error is $F_1 \cdot F_2 - \phi(E_1 * E_2)$. Then

$$F_1 \cdot F_2 - \phi(E_1 * E_2) = F_1F_2 - \{\phi(E_1) \cdot \phi(E_2)\} + kr^m$$

$$= F_1F_2 - (F_1 - e_1)(F_2 - e_2) - kr^m$$

$$= e_1F_2 + e_2F_1 - e_1e_2 - kr^m.$$

Chapter 7 shows the modulo reduction factor in radix complement addition (or subtraction) is +1, 0 or -1. The error in a sum or difference of elements of C_m^r is then $+r^m$, 0 or $-r^m$. On the IBM 1130 in C_{16}^2, this error is +32768, 0 or -32768, while in C_{32}^2 on the IBM/370, it is +4294967296, 0 or -4294967296. In these cases, any error introduced by addition or subtraction has twice the magnitude of the largest number in the system. Error in the arguments may increase this.

Multiplication of two radix complement numbers may give extremely large error. The greatest overflow (and largest error) for multiplication occurs with Nm*Nm. On the IBM/370, $N_{32}^2 = -2147483648$ (-2^{31}) and $N_{32}^2 * N_{32}^2 = 0$, with modulo reduction factor $k = -1073741824$ (-2^{30}). The error in $N_{32}^2 * N_{32}^2$ is thus 4,611,686,018,427,387,904 (2^{62}). This is about 5×10^{18}; the largest element of C_{32}^2 is approximately 2×10^9.

The error in an expression E will be some multiple of r^m when the operations in E are all additions, subtractions and multiplications. To see this, note that the error in a single element of C_m^r (an expression

with no operations) is zero. The error in E is obtained by repeated application of the theorems above, and will therefore be composed of sums, differences and products of multiples of r^m.

Division will not cause overflow error; it does not overflow. However, it may modify and propagate error from its operands. Note that error in this context refers to differences between integer arithmetic and radix complement arithmetic, and not to differences with arithmetic in the rational numbers. The quotient $5/9 = 0$ and has no error, because it is the same in C_m^r and Z. Comparison to the rational quotient $\frac{5}{9}$ would have little meaning; rational division is not comparable to integer or radix complement division.

Before stating the theorem for division, the following lemma may help. It represents a generalization of many results from Chapter 7.

Lemma 8.5

Let x,y,a and b be integers such that x/y and $(x + a)/(y + b)$ are defined. If $r = \text{rem}\{(x + a)/(y + b)\}$ and $d = \{[(x + a)/(y + b)]\cdot b + r - a\}$, then $x/y = (x + a)/(y + a) + f$, where

$$f = \begin{cases} d/y & \text{if } s(d) = s(x) \text{ or } x = 0 \text{ or } \text{rem}(d/y) = 0, \\ d/y - 1 & \text{if } s(d) \neq s(x) \text{ and } s(x) = s(y), \text{ or} \\ d/y + 1 & \text{if } s(d) \neq s(x) \text{ and } s(x) \neq s(y). \end{cases}$$

Proof: Let $q = (x + a)/(y + b)$. The primary division theorem gives $x + a = q\cdot(y + b) + r$, where $r = \text{rem}\{(x + a)/(y + b)\}$. Then $x = q\cdot y + (q\cdot b + r - a)$. With $d = q\cdot b + r - a$, the reduction theorem provides the conclusion.

Theorem 8.6

Let the expressions and errors be as given for Theorems 8.2, 8.3 and 8.4 and assume E_1/E_2 and F_1/F_2 are defined. If $r = \text{rem}(E_1/E_2)$ and $d = r + e_1 - e_2 \cdot \phi(E_1/E_2)$, then the error in E_1/E_2 is given by

$$\begin{cases} d/F_2 & \text{if } s(d) = s(F_1) \text{ or } F_1 = 0 \text{ or } \text{rem}(d/F_2) = 0, \\ d/F_2 - 1 & \text{if } s(d) \neq s(F_1) \text{ and } s(F_1) = s(F_2), \text{ or} \\ d/F_2 + 1 & \text{if } s(d) \neq s(F_1) \text{ and } s(F_1) \neq s(F_2). \end{cases}$$

Proof: The error is $F_1/F_2 - \phi(E_1/E_2)$. This becomes $F_1/F_2 - \phi(E_1)/\phi(E_2)$ and $F_1/F_2 - (F_1 - e_1)/(F_2 - e_2)$. Taking $x = F_1$, $y = F_2$, $a = -e_1$ and $b = -e_2$ in Lemma 8.5 will complete the proof.

The table below summarizes the results. No distinction is made here between X and $\phi(X)$, or Y and $\phi(Y)$.

Value in C_m^r	"Correct" Value in Z	Error
X	$X + a$	a
Y	$Y + b$	b
$X \oplus Y$	$\phi(X \oplus Y) + a + b - kr^m$ $(k = 1, 0, -1)$	$a + b - kr^m$
$X \ominus Y$	$\phi(X \ominus Y) + a - b - kr^m$ $(k = 1, 0, -1)$	$a - b - kr^m$
$X*Y$	$\phi(X*Y) + Xb + Ya - ab - kr^m$ (k an integer)	$Xa + Yb - ab - kr^m$
X/Y	$\phi(X/Y) + f$ (f given in Theorem 8.6)	f

These rules suggest an inductive process for determining the error in any finite expression over C_m^r. However, the conditions governing the choice of the value for k (or f) are sufficiently complicated to

make this impractical in general. Specific expressions may be evaluated for error on a case by case basis.

The next application will draw heavily on results in Chapter 7 to produce a representation in $Z \times Z \times Z$ of the rational numbers in mixed form.

CHAPTER 9

AN ALGEBRA FOR MIXED FORM RATIONALS

The rational numbers are usually defined as a set of ordered pairs of integers. In "mixed form," a rational number (a,b) or $\frac{a}{b}$ is expressed as $q + \frac{r}{b}$, where $q = a/b$ is an integer, and $\frac{r}{b} = \frac{rem(a/b)}{b}$ is a rational fraction having magnitude less than one and sign the same as $\frac{a}{b}$. This suggests representing the rational numbers, in mixed form, as a subset of the set of ordered triples of integers, $Z \times Z \times Z$. Such a representation for $\frac{a}{b}$ will require that one component be the integer part a/b, one the remainder $rem(a/b)$, and one the denominator b.

Many results from Chapter 7 will be used in developing this system of ordered triples. Because all arithmetic will be performed in the integers, the no-overflow condition will hold throughout. The theory in Chapter 7 for the no-overflow case applies directly to the integers.

The Ordered Triples

The following defines a map from a set of ordered pairs of integers (rational numbers) to the set of ordered triples of integers.

Definition 9.1

Let $div^*: Z \times \{Z - 0\} \to Z^3$ be defined for (a,b) in $Z \times \{Z - 0\}$ by $div^*(a,b) = (q,r,b)$, where $q = a/b$ and $r = rem(a/b)$. (rem is the remainder map defined in Chapter 7.)

The map is well-defined, because the quotient q and the remainder r associated with the primary quotient a/b are unique. It is also one-to-one. This is seen by noting that if $div^*(a',b') = div^*(a,b) = (q,r,b)$, then $b = b'$. The primary division algorithm shows that $a = qb + r$, so $a = a'$, and therefore, $(a',b') = (a,b)$. It must be carefully noted here that the $=$ symbol used with the pairs and triples represents the usual equivalence relation on sets of ordered tuples; two ordered n-tuples are equal if and only if their corresponding components are equal. This is not the same as the equivalence relation on $Z \times \{Z - 0\}$, denoted by $=_Q$, which produces the system of rational numbers.

The entire set of ordered triples of integers Z^3 will not be of interest, because many triples cannot be derived from an ordered pair using Definition 9.1. In particular, the primary quotient requires that the remainder r have the same sign as $qb + r$, and that $|r| < |b|$. Thus, attention will be restricted to those triples which lie in the range of div^*.

Definition 9.2

Let $D^* = div^*(Z \times \{Z - 0\})$.

D^* is the set of ordered triples which will make up the mixed form rational system. Before proceding, it might be helpful to formally characterize those triples in D^*.

Theorem 9.3

Let q, b and r be integers. Then $(q,r,b) \in D^*$ if and only if $s(r) = s(qb + r)$ and $|r| < |b|$, where s is the sign function

defined previously.

Proof: This follows directly from the definitions of div^* and the primary quotient. Uniqueness of the form $qb + r$ is used for the sufficiency.

A Rational System for D^*

From Definition 9.2 and earlier remarks, the map div^* is one-to-one onto the set of triples D^* in Z. Thus, this map may be used to create an algebraic structure in D^* which is a copy of the structure in $Z \times \{Z - 0\}$ called the rational number system. The definitions required in D^* are an equivalence relation, an addition, and a multiplication. The next three definitions form the backbone of the usual rational number system.

Definition Q.1

Let (a,b) and (c,d) be elements of $Z \times \{Z - 0\}$. Then

$$(a,b) =_Q (c,d) \text{ if and only if } ad = bc.$$

Definition Q.2

Let (a,b), (c,d) and (x,y) be elements of $Z \times \{Z - 0\}$. Then

$$(a,b) + (c,d) =_Q (x,y) \text{ if and only if } (x,y) =_Q (ad + bc, bd).$$

Definition Q.3

Let (a,b), (c,d) and (x,y) be elements of $Z \times \{Z - 0\}$. Then

$$(a,b) \cdot (c,d) =_Q (x,y) \text{ if and only if } (x,y) =_Q (ac,bd).$$

The map div^* is used to duplicate these definitions in D^*. Because div^* is one-to-one and onto, they are well defined.

Definition 9.4

Let (q,r,b) and (q',r',b') be elements of D^*, and let (a,b) and (a',b') be the elements of $Z \times \{Z - 0\}$ for which $div^*(a,b) = (q,r,b)$ and $div^*(a',b') = (q',r',b')$. Then define $(q,r,b) =_D (q',r',b')$ if and only if $(a,b) =_Q (a',b')$.

Note that the symbol $=_D$ is used for this relation on D^*; the nature of div^* makes $=_D$ an equivalence relation on D^*.

Definition 9.5

Let (q,r,b) and (q',r',b') be elements of D^*, with (a,b) and (a',b') the pre-images of (q,r,b) and (q',r',b'), respectively. Then

$$(q,r,b) + (q',r',b') = div^*[(a,b) + (a',b')].$$

The one-to-one nature of div^* ensures that this sum is well-defined. To show this, suppose that $(q,r,b) =_D (x,y,z)$ and $(q',r',b') =_D (x',y',z')$. Let (c,d) and (c',d') be the corresponding pairs for (x,y,z) and (x',y',z'). Then $(a,b) =_Q (c,d)$ and $(a',b') =_Q (c',d')$, so that

$$(a,b) + (a',b') =_Q (c,d) + (c',d').$$

Thus $$div^*[(a,b) + (a',b')] =_D div^*[(c,d) + (c',d')],$$

and $$(q,r,b) + (q',r',b') =_D (x,y,z) + (x',y',z').$$

Definition 9.6

Let (q,r,b) and (q',r',b') be elements of D^*, with (a,b) and (a',b') as before. Define $(q,r,b)\cdot(q',r',b') = div^*[(a,b)\cdot(a',b')]$.

The proof that this product is well-defined follows in the same way as that for the sum.

These definitions may now be characterized in terms of the elements of D^*, so it will not be necessary to use $Z \times \{Z - 0\}$ and div^* in working with D^*. This is done as follows.

Theorem 9.7

Let (q,r,b) and (q',r',b') be elements of D^*. Then $(q,r,b) =_D (q',r',b')$ if and only if $q = q'$ and $(r,b) =_Q (r',b')$.

Proof: Suppose that $(q,r,b) =_D (q',r',b')$. Then there are pairs (a,b) and (a',b') such that $div^*(a,b) = (q,r,b)$, $div^*(a',b') = (q',r',b')$, and $(a,b) =_Q (a',b')$. This last fact implies that $ab' = a'b$. From the definition of div^*,

$$a = qb + r \quad \text{and} \quad a' = q'b' + r,$$

so $$ab' = qbb' + rb' \quad \text{and} \quad a'b = q'b'b + r'b.$$

Since $ab' = a'b$, it follows that

(1) $$ab' = qbb' + rb' = q'bb' + r'b.$$

The primary quotient ensures that $|r| < |b|$, and $|r'| < |b'|$, so

(2) $$|rb'| < |bb'| \quad \text{and} \quad |r'b| < |bb'|.$$

Now, if either r or r' is zero, equation (1) above shows that the integer ab' is divisible by bb'. The magnitude restrictions on rb' and $r'b$ then imply that both r and r' must be zero. This means that $(r,b) =_Q (r',b')$. It also means $qbb' = q'bb'$, which implies $q = q'$, completing this part of the proof for the case when r or r' is zero.

Suppose neither r nor r' is zero. The magnitude restrictions in (2) still apply. In addition, neither a nor a' can be zero, and the primary quotient asserts that s(r) = s(a) and s(r') = s(a'), where s is the sign function. Thus, s(rb') = s(ab') and s(r'b) = s(a'b). Because ab' = a'b, it follows that s(rb') = s(r'b) = s(ab'). Under these conditions, it must be true that equations

$$ab' = qbb' + rb' \qquad \text{and} \qquad ab' = q'bb' + r'b$$

both represent the quotient (ab')/(bb'). The uniqueness property (Theorem 3.3) then says that q = q' and rb' = r'b. This latter provides $(r,b) =_Q (r',b')$, so this part of the proof is complete.

Now, suppose q = q' and $(r,b) =_Q (r',b')$ for elements (q,r,b) and (q',r',b') of D^*. As before, there are pairs (a,b) and (a',b') such that a = qb + r and a' = q'b' + r'. This means ab' = qbb' + rb' and a'b = q'bb' + r'b. Since q = q' gives qbb' = q'bb', and $(r,b) =_Q (r',b')$ gives rb' = r'b, it follows that ab' = a'b. Thus, $(a,b) =_Q (a',b')$, which implies $(q,r,b) =_D (q',r',b')$ and completes the proof of Theorem 9.7.

This theorem provides the first clear indication that the system D^* is the rational number system in mixed form.

Characterization of addition and multiplication is done in two steps. First, the resulting triple is expressed using components of ordered pairs; the ordered pairs correspond under div^* to the ordered triple arguments of the sum or product. The results of Chapter 7 are then used to remove the ordered pairs from the picture.

Lemma 9.8

Let (q,r,b) and (q',r',b') be elements of D^*, with corresponding pairs (a,b) and (a',b'), respectively. Then $(q,r,b) + (q',r',b') = (x,y,z)$, where

$$x = (ab' + a'b)/(bb'),$$
$$y = ab' + a'b - xbb', \quad \text{and}$$
$$z = bb'.$$

Proof: This follows from the definition of div^*. The sum gives

$$(q,r,b) + (q',r',b') = \text{div}^*[(a,b) + (a',b')].$$

Since $(a,b) + (a',b') = (ab' + a'b, bb')$, application of div^* to this pair yields the conclusion.

Theorem 9.9

Let (q,r,b) and (q',r',b') be elements of D^*. Then

$$(q,r,b) + (q',r',b') = (x,y,z),$$

where

$$x = q + q' + f,$$
$$y = rb' + r'b - fbb', \quad \text{and}$$
$$z = bb',$$

with f the integer function defined in Corollary 7.22. (Corollary 7.22 is the no-overflow version of Theorem 7.21.)

Proof: Lemma 9.8 gives $x = (ab' + a'b)/(bb')$. Corollary 7.22 shows that $(ab' + a'b)/(bb') = (a/b) + (a'/b') + f$, where f is the integer function defined in that corollary. Since $q = a/b$ and $q' = a'/b'$, it follows that $x = q + q' + f$. The expression for f

in Theorem 7.21 contains references to a and a'. These may be eliminated using $a = qb + r$ and $a' = q'b' + r'$.

Expanding $y = ab' + a'b - xbb'$ using the expression for x, with $a = qb + r$ and $a' = q'b' + r'$, gives

$$y = (qb + r)b' + (q'b' + r')b - (q + q' + f)bb'.$$

Simplifying this shows $y = rb' + r'b - fbb'$.

The expression for z comes directly from Lemma 9.8.

As noted in Chapter 7, the function f may take any of the values +1, 0 or -1.

Lemma 9.10

Let (q,r,b) and (q',r',b') be elements of D^*, with corresponding pairs (a,b) and (a',b'), respectively. Then,

$$(q,r,b)\cdot(q',r',b') = (x,y,z),$$

where

$$x = (aa')/(bb'),$$
$$y = aa' - xbb', \quad \text{and}$$
$$z = bb'.$$

Proof: This follows again from the definition of div^*, the definition of the sum of triples, and the fact that $(a,b)\cdot(a',b') = (aa',bb')$.

Theorem 9.11

Let (q,r,b) and (q',r',b') be elements of D^*. Then

$$(q,r,b)\cdot(q',r',b') = (x,y,z),$$

where

$$x = qq' + f,$$
$$y = qbr' + q'b'r + rr' - fbb', \quad \text{and}$$
$$z = bb',$$

with f the integer function defined in Corollary 7.25. (Corollary 7.25 is the no-overflow version of Theorem 7.24.)

Proof: The lemma shows $x = (aa')/(bb')$, and Corollary 7.25 gives $(aa')/(bb') = (a/b)\cdot(a'/b') + f$, where f is the integer function defined in the corollary. Because $q = a/b$ and $q' = a'/b'$, it follows that $x = qq' + f$. The expression for y is derived as in Theorem 9.9, and that for z from Lemma 9.10. References to a and a' in f may be eliminated using $a = qb + r$ and $a' = q'b' + r'$.

The Mixed Form Rationals

div^* is a bimorphism for both addition and multiplication. This follows from the definitions of the relation $=_D$ and operations $+$ and $\cdot$ in D^*.

Theorem 9.12

The set of triples $D^* \subset Z^3$ is isomorphic to the set of rational numbers Q as a field. div^* is the isomorphism.

The following examples demonstrate some basic properties of the field D^*.

Example 9.13

The zero and unity elements are preserved by the isomorphism div^*. Since the rational number 0 is represented by $(0,b)$, with $b \neq 0$, the zero in D^* is represented by $(0,0,b)$, for $b \neq 0$. Any triple equivalent to this under $=_D$ also represents 0. The number 1 is expressed as (a,a) in Q, if $a \neq 0$. This means $(1,0,a)$, with $a \neq 0$, is the unit element in D^*. Its negative -1 is given by $(-1,0,a)$.

Example 9.14

The additive inverse of an arbitrary element (q,r,b) of D^* may be found using div^*. If (a,b) is the ordered pair corresponding to (q,r,b), then $(-a,b)$ corresponds to its negative. Thus $-(q,r,b) =_D (-q,-r,b)$.

Example 9.15

The multiplicative inverse of (q,r,b) has a more complicated expression. If $div^*(a,b) = (q,r,b)$, with $a \neq 0$, then $div^*(b,a) =_D (q,r,b)^{-1}$. This means

$$(q,r,b)^{-1} =_D (b/(qb+r),\ rem\{b/(qb+r)\},qb+r),$$

where rem is the remainder map. This may be simplified by using three cases.

If $|q| > 1$, or if $|q| = 1$ and $r \neq 0$, then $|a| > |b|$. Since $a = qb + r$, this means $b/(qb+r) = 0$ and $rem\{b/(qb+r)\} = b$. If $|q| = 1$ and $r = 0$, then $a/b = b/a$, so $b/(qb+r) = q$ and $rem\{b/(qb+r)\} = 0$. Third, if $q = 0$, then $qb + r = r$, so $b/(qb+r) = b/r$ and $rem\{b/(qb+r)\} = 0$. Summarizing these,

$$(q,r,b)^{-1} =_D \begin{cases} (0,b,qb+r) & \text{if } |q| > 1, \text{ or } |q| = 1 \text{ and } r \neq 0, \\ (q,0,qb+r) & \text{if } |q| = 1 \text{ and } r = 0, \text{ or} \\ (b/r,rem(b/r),r) & \text{if } q = 0. \end{cases}$$

Extension to C_m^r

The construction of D^* uses theory developed for the radix complement system and its division operation. The radix complement

system was not, however, directly involved; D^* was based on Z, not C_m^r. Furthermore, an attempt to duplicate this for a system of radix complement triples fails.

The failure happens because C_m^r is modular. It appears when overflow occurs; in the absence of overflow, triples from C_m^r behave exactly like triples from Z. Zero divisors exist in C_m^r, so the addition and multiplication operations provided by Theorems 9.9 and 9.11 are not always closed. In C_{32}^2 on the IBM/370, the sum of (0,1,262144) and (0,1,16384) is not defined. In Theorem 9.9, division by 262144 * 16384 must be defined, but 262144 * 16384 = 0. For the same reason, the product of (0,1,262144) and (0,1,16384) given by Theorem 9.11 is undefined.

A more subtle failure in the equivalence relation of Theorem 9.7 is illustrated in the following example.

Example 9.17

Consider $C_6^2 = \{-32, -31, \ldots, 30, 31\}$. In this system, addition and multiplication are performed modulo 64. Take triples $x = (1,3,7)$, $y = (1,6,14)$ and $z = (1,1,13)$ in $C_6^2 \times C_6^2 \times C_6^2$, and let $=_C$ be a relation defined as in Theorem 9.7. This means $(Q,R,B) =_C (Q',R',B')$ if and only if $Q = Q'$ and $R*B' = R'*B$, where overflow may occur because C_6^2 is modular.

Now $x =_C y$ because $3*14 = -22 = 7*6 \pmod{64}$. Since $6*13 = 14 = 1*14$, it also follows that $y =_C z$. However, $3*13 = -25 \neq 7*1$, so $x \neq_C z$. Transitivity fails for $=_C$ because $x =_C y$ and $y =_C z$ with $x \neq_C z$. Therefore, $=_C$ is not an equivalence relation on C_6^2.

It may be interesting to note that each triple used in Example 9.17 is derived from an element of $C_6^2 \times \{C_6^2 - 0\}$ using a map equivalent to div^*. The pairs and corresponding triples are

$$(10,7) \rightarrow (1,3,7),$$
$$(20,14) \rightarrow (1,6,14) \quad \text{and}$$
$$(14,13) \rightarrow (1,1,13).$$

These examples indicate that a system of triples based on the radix complement system will have to differ substantially from the D^* system of integer triples. Meaningful definitions may not be possible in the case of overflow. Examination of this situation could nevertheless prove valuable. A system of ordered triples from the radix complement system which simulates the rational numbers might offer advantages over the usual computer floating point arithmetic. This extension will not, however, be pursued here.

CHAPTER 10

SUMMARY AND CONCLUSIONS

The radix complement system used for integer arithmetic on most modern computers is described in any text dealing with computer architecture or computer arithmetic. However, these descriptions are usually limited to an explanation of how the arithmetic is performed, and a hint that the computer arithmetic, within constraints, appears similar to arithmetic in the ring of integers. Computer multiplication and division are generally described in terms of the corresponding integer operations. The modular structure of the radix complement system is mentioned by Rao [13] and Tremblay [15]; Rao sketches a proof. Because of the growing importance of computer integer systems, a deeper study of them was undertaken as the subject of this dissertation.

Believed by many users to represent integers (from the mathematical system of integers), computer integers are, in fact, a modular system isomorphic to the ring of integers modulo k for some k. The value of k depends both on the radix (base) and on the number of digits with which elements are represented. Most common radices are small powers of 2 (2, 8 or 16). The number of digits used may depend on both the computer word size, and the programming (multiple precision). If numbers are represented using m digits in radix r, the radix

complement system is equivalent to the integers modulo r^m.

Although it is a modular system, the primary use of the radix complement system is to simulate operations from the mathematical ring of integers. Because of this, the class representatives used in the modular system are not the usual non-negative values, but include a range of positive and negative numbers (about the same number of each) which straddles zero. If a condition called "overflow" does not occur, arithmetic in this system exactly simulates that of the ring of integers. With overflow, the simulation breaks down because the modular structure comes into play; high-order integer digits must be truncated to give modular closure. Most users consider obtaining a sum of positive squares which is negative to be an error; it is nothing more than the proper operation of an integer modular system.

Division in the radix complement system, introduced in Chapter 3, duplicates the standard "long division" of the integers for a restricted set of arguments. This long division closely resembles Euclidean division in the ring of integers. Computer division is defined using integer division with the signed integer representatives of the modular classes from the radix complement system. The primary division theorem for the ring of integers, an important tool for later use, provides a unique form for the quotient and remainder in division. While the division theorem carries over to the radix complement system, the uniqueness property fails there because that system is modular.

Computer integer division is not a modular operation. This is shown in Chapter 4 where computer division is studied in greater depth. The modular homomorphism from the ring of integers onto the radix comple-

ment system is not a morphism for division, and, significantly, cannot be extended to become such. A comparison of computer division and modular division is also made. The radix complement system is modular, so non-trivial multiplicative inverses exist; 3*(-1431655765) = 1 on the IBM 370 FORTRAN. The modular quotient, defined using these inverses, is generally distinct from the computer quotient. Finally, the elements of the radix complement system, together with the division operation, are considered as an algebraic system; few of the classical algebraic axioms are satisfied, but new insight into this division is gained.

An order relation on the ring of integers is a very useful feature. Two integers may be compared, and the smaller or larger identified. The usual algebra of inequalities applies, so that equations of inequality may be solved. The radix complement system, however, is modular and cannot have an order. It is not even possible to compare elements using subtraction, because overflow will invalidate the conclusion. This last fact is known by computer manufacturers, but not by most current users. The possibility of extending some form of "quasi-order" to the radix complement system is discussed in Chapter 5.

Restricting the integer order relations to the signed values of the modular classes provides a useful "quasi-order" for the radix complement system. This quasi-order gives the comparison relations of Chapter 5. These c-relations can compare computer integers, but do not permit the usual algebra of inequalities. The comparison of elements can be implemented either by a special comparison operator, or by subtraction with a logical correction for overflow.

Chapter 6 extends this work to provide a limited algebra of

inequalities. When overflow is absent, this system of "quasi-inequalities" (c-inequalities) exhibits the same properties as do inequalities in the ring of integers. With overflow, the results are considerably different. If the amount of overflow is minimal, as with addition or subtraction, then manipulation of these c-inequalities is possible, although complicated. If overflow is more than minimal, as may occur in multiplication, little can be done. Division does not involve overflow, so for this operation, c-inequalities mimic integer inequalities. Because of the importance of overflow, it is quantized by means of the modulo reduction factor. Significant in its own right, this factor is an important component in much of the work of succeeding chapters, and permits us to derive several new theorems related to the actual behavior of computer integer arithmetic.

The focus returns in Chapter 7 to computer integer division. Division is considered in relation to the ring operations addition, subtraction and multiplication. In either the radix complement system or the ring of integers, division is a distinct operation. It is not related to multiplication in the way that division in a field is related to field multiplication. The law of cancellation for products holds if overflow does not occur. Other field-related rules involving multiplication and division fail unless the division is exact. One use of computer division is to test for divisibility; the denominator (exactly) divides the numerator if and only if the computer remainder is zero.

A significant step is made in Chapter 7 by considering, in the radix complement system, rules of fractions for a field. This study uses the primary division theorem of Chapter 3 to evaluate computer

integer division in the presence of other radix complement operations. If overflow does not occur during evaluation of expressions, a rule of fractions for field may be true for parts of the computer integer system. However, that field property may also fail badly. This last is the case when overflow occurs during evaluation. Of particular note, this failure is quantized in both the overflow and non-overflow cases. The quantization for overflow strongly involves the modulo reduction factor of Chapter 6.

The occurrance of overflow is usually considered an error by computer users. This "error," a natural characteristic of the modular system, is considered in Chapter 8. Overflow in one of the operations addition, subtraction or multiplication produces error that is a multiple of the modular base of the system. If division is involved, further error is not introduced, but any error present may be modified and passed on. Techniques developed in Chapter 7 for dealing with division in relation to the other operations allow a quantization of this error propagation. This type of error is truly catastrophic. The magnitudes involved are often greater than the largest values which may be represented in the system.

The modified rules of "fractions," developed in Chapter 7 for the radix complement system, are used in Chapter 9 to devise a non-standard representation for the rational number field in a mixed number (integer plus fraction) form. The fraction part is given by a pair of integers, so the rational number is represented by an ordered triple of integers. The rules for arithmetic in this field are derived. If the ordered triple is taken from the radix complement system, a computer

representation of rational numbers is possible.

Although our problem was restricted to computer integer systems, a system for simulating rational arithmetic, based on ordered triples of computer integers, could be useful. It would resemble the triple system based on the integers, but allowance would have to be made for overflow; the very real problem of zero divisors and zero denominators in fractions would have to be solved. As a rational-like system, its range of values would be approximately the same as the range of signed values available in the computer integer system. This would be considerably less than the dynamic range in a floating point system. However, the distribution of values would be much more uniform than in either the floating point system or a rational system based on ordered pairs of computer integers. The arithmetic for such a system of triples would be complicated.

This is a question for the future. It is not one of the problems examined by this dissertation and will not be persued here. It is hoped that the results and techniques presented here will prove useful in the consideration of other computer arithmetic systems.

BIBLIOGRAPHY

1. Amdahl, G. M., G. A. Blaauw, F. P. Brooks, Jr. "Architecture of the IBM System/360," IBM Journal of Research and Development, v.8, no.2, April 1964, pp.87-101.

2. Cardenas, A. F., L. Presser, M. A. Marin, eds. Computer Science. New York: Wiley-Interscience, 1972.

3. Chinal, J. Design Methods for Digital Systems. New York: Springer-Verlag, 1973.

4. Chu, Yaohan. Computer Organization and Microprogramming. New Jersey: Prentice-Hall, 1972.

5. Davenport, H. The Higher Arithmetic. New York: Harper, 1952.

6. Dudley, Underwood. Elementary Number Theory. San Francisco: W. H. Freeman, 1969.

7. Ehrman, J. R. "'Logical' Arithmetic on Computers with Two's Complement Binary Arithmetic," Communications of the ACM, v.11, 1968, pp.517-520.

8. Falkoff, A. D., K. E. Iverson, E. H. Sussenguth. "A Formal Description of System/360," IBM Systems Journal, v.3, no.3, 1964, pp.198-262.

9. Flores, Ivan. Computer Organization. New Jersey: Prentice-Hall, 1969.

10. Gear, C. William. Computer Organization and Programming. New York: McGraw-Hill, 1969.

11. IBM System/370 Principles of Operation. Fourth Ed. (January 1973) IBM System Products Division.

12. Knuth, Donald E. The Art of Computer Programming. v.2. Semi-numerical Algorithms. Reading: Addison-Wesley, 1969.

13. Rao, T. R. N. Error Coding for Arithmetic Processes. New York: Academic Press, 1974.

14. Struble, George. Assembler Language Programming: The IBM System/360. Reading,Mass.: Addison-Wesley, 1969.

15. Tremblay, J. P., R. Manohar. Discrete Mathematical Structures with Applications to Computer Science. New York: McGraw-Hill, 1975.

16. User's Manual, Programmer's Reference, microNOVA Computers. Data General Corporation, 1976.

17. Weiss, Eric. A., ed. Computer Usage/Fundamentals. New York: McGraw-Hill, 1969.

18. Wilkinson, J. H. Rounding Errors in Algebraic Processes. New Jersey: Prentice-Hall, 1964.