Science World Journal Vol 6 (No 1) 2011
www.scienceworldjournal.org
ISSN 1597-6343

Short Communication Report

COMPUTER REPRESENTATION OF MULTISETS

*Tella, Y. & Daniel, S.

Department of Mathematical Sciences
Kaduna State University, Kaduna.
*yot2002ng@yahoo.com

INTRODUCTION

A Multiset (mset for short) is an unordered collection of objects in which repetition of elements is significant. We confront a number of situations in life when we have to deal with collections of elements in which duplicates are significant. An example may be cited to prove this point. While handling a collection of employee's ages or details of salary in a company, we need to handle entries bearing repetitions and consequently our interest may be diverted to the distribution of elements. In such situations the classical definition of sets proves inadequate for the situation presented. Thus from the practical point of view, msets are very useful structures as they arise in many areas of mathematics and computer science. The prime factorization of integers $n > 0$ is an mset whose elements are primes. Infact repeated observations in statistical samples, repeated hydrogen atoms in a water molecule, H_2O, e.t.c. need to be counted for attaining adequacy and exactness. Every monic polynomial $f(z)$ over complex numbers corresponds in a natural way to the mset of its roots. Other examples of msets include the zeros and poles of meromorphic functions, invariant of matrices in a canonical for etc.

There are various ways to represent sets using a Computer. One method is to store the elements of the set in an unordered fashion. If this is done the operation of computing the union, intersection, complement and difference of two sets would be time-consuming because each of these operations would require a large amount of searching for elements. However, a method of storing elements using arbitrary ordering of elements of the universal set makes computing combination of sets easy. In this method, the universal set μ is assumed finite and reasonable in size so that the number of elements of μ is not larger than the memory size of the Computer being used. The method specifies an arbitrary ordering of the elements of μ, for instance $\{a_1, a_2, ..., a_n\}$ represents the subset A of μ with the bit string of length n, where the i^{th} bit in this string is 1 if $a_i \in A$ and 0 if $a_i \notin A$ (Rosen, 1999)..

In this paper, we put forward similar representations for easy computation of msets. First, we started with some fundamentals of msets and then some of its operations. We thendefined our equivalent Computer representation of msets and adopt the use of bitwise Boolean operations to define an equivalent representation for Union, intersection, Complement and difference operations on msets.

Preliminaries

An mset is an unordered collection of objects (called its elements) in which unlike standard (Cantorian) sets, repetition of elements is significant. From the classical point of view, $\{a,b,b,c\} = \{a,b,c\}$. However $\{a,b,b,c\} \neq \{a,b,c\}$ where repetitions or duplicates are significant.

Msets are denoted by the commonly used function symbols $f, g, h, ...$. For elements $x \in D$, $A(x)$ is called the *multiplicity* of x in A and the sum of all the multiplicities of an mset is called its *cardinality*. It follows by definition that $A(x) > 0 \ \ \forall x \in A$.

Mset M is *included* (or submultiset or submset for short) in mset N (denoted $M \subseteq N$) if and only if $M(x) \leq N(x)$ for all $x \in D$.

The *union* of two msets M and N is the mset $M \cup N$ such that $(M \cup N)(x) = \max\{M(x), N(x)\}$.

The *intersection* of two msets M and N is the mset $M \cap N$ such that $(M \cap N)(x) = \min\{M(x), N(x)\}$

The *difference* of two msets M and N is the mset $M - N$ such that

$$(M - N)(x) = \max\{M(x) - N(x), 0\} = M(x) \dot{-} N(x)$$

where

$$M(x) \dot{-} N(x) = \begin{cases} M(x) - N(x) \text{ if } M(x) > N(x) \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases}$$

Let M be an mset. For any mset N such that $M \subseteq N$, the complement of M relative to N denoted $\overline{M}$ is defined: $\overline{M} = N - M$ (Blizard,1989; Petrovsky, 2004).

Bitwise Boolean operations.

At their lowest level digital Computers handles only binary signals, represented with the symbols 0 and 1. The most elementary circuits that combine those signals are called gates. These gates are: OR, AND and NOT. Their outputs can be expressed as a function of their inputs by the logic tables. Using the logic symbols $\vee$, $\wedge$ and $\neg$ for operations representing OR, AND and NOT respectively and $0,1$ for F, T respectively, we have the following Bitwise Boolean operation tables.

x_1	x_2	$x_1 \vee x_2$
1	1	1
1	0	1
0	1	1
0	0	0

(a)

x_1	x_2	$x_1 \wedge x_2$
1	1	1
1	0	0
0	1	0
0	0	0

(b)

x_1	$\neg x_1$
1	0
0	1

(c)

Bit string for submset

We put forward a binary representation of an msubset relative to its parent mset based on the following assumptions:
(i) Each occurrence of an element in an mset is distinct irrespective of its repetition.
(ii) The parent mset of an submset is finite.
(iii) An arbitrary ordering on the parent mset.

Our method takes an arbitrary msubset $M = \{a_1, a_2, ..., a_n\}$ of the parent mset N with the bit string of length n, where the i^{th} bit in this string is 1 if $a_i \in M$ and 0 if $a_i \notin M$.

For example Let $N = \{a, a, a, b, b, b, c, c, c, c, c\}$. The binary (Bitwise representation) of the submset $M = \{b, b, c, c, c\}, P = \{c, c, a, b\}$ are 00011011100 and 00110011000 respectively.

Bit string for mset operations

To obtain the bit string for the union, intersection, Complementation and difference of submsets we perform bitwise Boolean operations on the bit strings representing the msets. The bit in the i^{th} position of the bit string of the mset union is 1 if either of the bits in the i^{th} position in the two strings is 1 (or both are 1), and is 0 when both bits are 0. Hence, the bit string for the union is the bitwise OR of the bit strings for the two msets.

The bit in the i^{th} position of the bit string of the mset intersection is 1 when the bits in the corresponding position in the two strings are both 1, and is 0 when either of the two bits is 0 (or both). Hence, the bit string for the intersection is the bitwise AND of the bit strings for the two msets.

To find the bit string for the complement of an mset from the bit string for that mset, we simply change each 1 to a 0 and each 0 to 1, because for each distinct occurrence, $x \in M$ if and only if the distinct occurrence $x \notin \overline{M}$ for any mset M .

Note that this operation corresponds to taking the negation of each bit when we associate a bit with a true value - with 1 representing true and 0 representing false.

For the difference of two msets, we compare the bit string representations of the two msets and uphold 1 where we have 1 and 0 bit strings in the comparisms. Note that 1 is a bit string for an occurrence in the first mset in the order of the difference operation.

Example: Let $N = \{a, a, a, b, b, b, c, c, c, c, c\}$ and $M = \{b, b, c, c, c\}, P = \{c, c, a, b\}$ be the submsets of N .

The bit strings for M and P are 00011011100 and 00110011000 respectively.

Therefore $M \cup P$ bit strings are:
$00011011100 \vee 00110011000 = 00111011100$
representing an mset $\{a, b, b, c, c, c\}$, $M \cap P$ bit strings are:
$00011011100 \wedge 00110011000 = 00010011000$
representing the mset $\{b, c, c\}$.

$\overline{M}$ bit strings are: 11100100011 representing the mset $\{a, a, a, b, c, c\}$.

The bit strings for $M - P$ are:
$00011011100 \Theta 00110011000 = 00001000100$
representing the mset $\{b, c\}$.

REFERENCES

Blizard, W. (1989), Multiset theory. Notre Dame Journal of formal logic, Vol. 30: 36-66.

Petrovsky A. B. (2004), Multi-attribute classification of credit cardholders: Multiset approach, MCDM, Whistler B.C Canada August 6-11.

Rosen, K. H. (1999). *Discrete mathematics and its applications,* McGraw-Hill