Massachusetts Institute of Technology
Cambridge, Massachusetts
Project MAC

Memorandum MAC-M-246
June 14, 1965

89A

Computer Experiments in Finite Algebra

by W. D. Maurer

The experiments described here concern an initial design for a computer system specifically for the handling of finite groups, rings, fields, semigroups, and vector spaces. The usefulness of such a system was discussed in (1). The system has been coded in MAD, with certain subroutines in FAP, for the IBM 7094, and is designed to operate in a time-sharing environment.

The initial design described below has certain inherent limitations. Some of these have to do with the design of the FORTRAN Monitor System (FMS), in which this system is embedded. The limitations are as follows:

(1) The system consists of a main routine, PHRASE, which accepts commands from the user's console, and a collection of 80 subroutines, each one of which is coded separately (except for a few which are grouped into modules, such as DASS, in which they exist as entry points) which carry out the 80 allowable functions. In order to insert a new function, the system programmer must: (a) write the function and compile it; (b) change the main routine, PHRASE, so that it contains a reference to the new routine in its transfer vector; (c) change one of the load files used by the loader (CTEST3 - June 1965) to include a reference to the new routine; (d) reload the entire set of routines and proceed.

A better design would involve keeping the subroutines on disk in relocatable format and loading each one as it is needed. In order to do this, we need a load routine which can be called by PHRASE, and which remains in core with PHRASE, which can load a given routine at object time, make its transfer vector references, and allocate storage for it.

(2) The underlying mechanism of the system does not incorporate a true list processor. All tables are logically considered as a set of single and double arrays which are kept one after the other in a large array (MEMORY). The name of each array, its index in MEMORY, and a two-way list giving alternate names for the same array are kept in the high end of MEMORY. A double array, one of whose dimensions is not known in advance, can be generated just above the end of the currently used space in MEMORY. More than one such array, however, cannote be generated.

Consideration will be given both to SLIP and to the AED free storage routines as a list processing medium for a subsequent design. With the arrival of the new file input-output system, these may be referenced without physically copying them into the same files in which the routines for the algebra system are contained.

(3) The system does not contain any general mechanisms for defining mathematical functions. Each one must be defined separately, and the definer must at all times consider how the algebra system is designed.

A useful component of a future design for an algebra system would be an inferential compiler. This is a routine which accepts, as source statements, mathematical statements (such as definitions, theorems, lemmas, facts) and which produces routines as output which verify or disprove a given theorem for a given particular case, by simple computation, or in general, by the methods of inferential analysis.

Use of the ALGEBRA System

Each routine is called by a calling phrase, which is an English sentence terminated by a period. Parameters in a calling phrase (which may be replaced by the names of Cayley tables, subsets, maps, or constants as appropriate) are represented by the symbols $1 through $9. The routines are requested, one by one, from the console. Upon entry to ALGBRA, the program types

```
*
```

The user should type the calling phrase of some routine, with parameters which he specifies. For example (using the first calling phrase listed below) he might type INPUT CAYLEY TABLE G1, ORDER 8. (The period is essential.) The program will ask further questions about the structure of G1, and will exit to the calling phrase routine with a copy of the Cayley table G1 in memory.

If a calling phrase is mistyped, ALGBRA will print

```
?
```

followed by another asterisk as above. Normally, ALGBRA executes the called routine and returns to type another asterisk.

A parameter may consist of a string of alphanumeric symbols from one to six characters long. The routines and their calling phrases are listed below.

IPCT INPUT CAYLEY TABLE $1, ORDER $2.

This routine allows a Cayley table to be input to ALGBRA from the console. IPCT and the next six routines (IPST, IPSC, IPMA, IPMD, IPSM, IPCO) give their own further directions.

IPST INPUT SUBSET $1.

This allows a subset to be input from the console. A subset of a Cayley table is given within the computer by a table containing non-zeros (usually ones) in each position corresponding to an element of the subset. A subset is typed in, under instructions from the console, by enumerating its elements.

IPSC INPUT SUBSET $1 OF $2.

This routine is the same as above, except that the specification of the containing set (by the second parameter) removes the necessity for the routine to ask for its cardinality (in order to allocate storage).

IPMA INPUT MAP $1.

This allows a map to be input by the user. A map is specified by enumerating the image of each element.

IPMD INPUT MAP $1 WITH DOMAIN $2.

This routine is the same as above, except that the specification of the domain of the map removes the necessity for the routine to ask for its cardinality (in order to allocate storage).

IPSM INPUT SET $1 OF $2 MAPS.

The second parameter of this routine specifies a number of maps in a set of maps to be input. A set of maps may be, e. g., a set of permutations; it is represented within the computer by a rectangular array.

IPCO INPUT CONSTANT $1.

This allows a constant to be input and given a name. This name may then be used by other routines.

<u>QUCT</u> PRINT CAYLEY TABLE $1.

The Cayley table given by the argument is displayed on the console.

<u>QUST</u> PRINT SUBSET $1.

The subset given by the argument is displayed on the console, together with a count of its elements.

<u>QUMA</u> PRINT MAP $1.

The map given by the argument is displayed on the console.

<u>QUSM</u> PRINT SET OF MAPS $1.

The set of maps given by the argument is displayed (as a rectangular array) on the console.

<u>QUCO</u> PRINT CONSTANT $1.

The constant whose name is given by the parameter is displayed on the console. Often a routine will return a constant (e. g., a number of subgroups) which the user should know in order to proceed.

<u>ERTA</u> ERASE TABLE $1.

The table, of whatever type (Cayley table, subset, map, set of maps) whose name is given by the parameter is erased. This provides more space in the computer for other tables. (Note: ERTA and the next routine, ERCO, erase quantities in core by setting the corresponding names to zero. The space does not actually become available until the tables are collapsed by means of COLAPS.)

<u>ERCO</u> ERASE CONSTANT $1.

The constant whose name is given by the parameter is erased.

RNTA   TABLE $2 IS $1

The table, of whatever type (Cayley table, subset, map, set of maps) whose name is given by the first parameter is renamed. That is, it now has two names, given by the two parameters. The old name is not deleted.

RNCO   CONSTANT $2 IS $1.

The constant whose name is given in the first parameter is given another name by the second parameter.

COLAPS   COLLAPSE TABLES.

The internal tables of ALGBRA are collapsed. Zero entries in the tables, corresponding to deleted constants, maps, or other objects, are eliminated.

GETOUT   QUIT.

This allows the user to quit without pushing the break button.

ALCT   ALTERNATING GROUP $2 ON $1 LETTERS.

This routine generates the Cayley table of the alternating group on the given number of letters. The name specified in the second parameter is given to this group.

SYCT   SYMMETRIC GROUP $2 ON $1 LETTERS.

This routine generates the Cayley table of the symmetric group on the given number of letters. The name specified in the second parameter is given to this group.

CYCT   CYCLIC GROUP $2, ORDER $1.

This routine generates the Cayley table of the cyclic group of the given order. The name specified in the second parameter is given to this group.

DHCT   DIHEDRAL GROUP $2 ON $1 VERTICES.

This routine generates the Cayley table of the dihedral group on the given number of vertices (the order of this group is twice the number of vertices). The name specified in the second parameter is given to this group.

LACT   LEFT ZERO SEMIGROUP $2, ORDER $1.

A left zero semigroup is an arbitrary finite set under multiplication given by xy = x. This routine generates the Cayley table of the left zero semigroup of the given order. The name specified in the second parameter is given to this semigroup.

RACT   RIGHT ZERO SEMIGROUP $2, ORDER $1.

A right zero semigroup is an arbitrary finite set under multiplication given by xy = y. This routine generates the Cayley table of the right zero semigroup of the given order. The name specified in the second parameter is given to this semigroup.

NACT   NULL SEMIGROUP $2, ORDER $1.

A null semigroup is an arbitrary finite set under multiplication given by xy = z, for some null element z. This routine generates the Cayley table of the null semigroup of the given order. The name specified in the second parameter is given to this semigroup.

CTDP   DIRECT PRODUCT $3 OF $1 AND $2.

The Cayley tables given in the first and second parameters are assumed to represent semigroups S1 and S2. The Cayley table of the direct product S1 x S2 is constructed and given the name specified in the third parameter.

DASS   GENERATE SET $2 OF ALL SUBSEMIGROUPS OF $1.

This routine determines all subsemigroups of the semigroup whose Cayley table is given by the first parameter. Each subsemigroup is produced as a subset (not as a separate Cayley table, although it may be brought to this form by the routine CTST described below). The subsets are grouped into a set of maps (i. e. subsets treated as maps). Individual subsets can be retrieved by the routine MASM (described below) and then printed as subsets; or the entire set of subsets can be printed by the routine DUSS (described below). Note: DASS and the following four routines, DALI, DARI, DATI, and DANS, determine subsets of a semigroup which satisfy certain properties. Except for the specific property involved, the characteristics of these routines are the same.

DALI GENERATE SET $2 OF ALL LEFT IDEALS OF $1.

This routine determines all left ideals of the semigroup whose Cayley table is given in the first parameter. The second parameter names the resulting set of left ideals.

DARI GENERATE SET $2 OF ALL RIGHT IDEALS OF $1.

This routine determines all right ideals of the semigroup whose Cayley table is given in the first parameter. The second parameter names the resulting set of right ideals.

DATI GENERATE SET $2 OF ALL TWO-SIDED IDEALS OF $1.

This routine determines all two-sided' ideals of the semigroup whose Cayley table is given in the first parameter. The second parameter names the resulting set of two-sided ideals.

DANS GENERATE SET $2 OF ALL NORMAL SUBGROUPS OF $1.

This routine determines all normal subgroups of the group whose Cayley table is given in the first parameter. The second parameter names the resulting set of normal subgroups.

DMSS GENERATE SET $2 OF ALL MAXIMAL SUBSEMIGROUPS OF $1.

This routine is the same as DASS excpet that only the maximal subsemigroups are evaluated. (In this routine and the next four routines, DMLI, DMRI, DMTI, and DMNS, "maximal" is short for "maximal proper"; in other words, a semigroup is never taken to be a maximal subset -- of any kind -- of itself.)

DMLI GENERATE SET $2 OF ALL MAXIMAL LEFT IDEALS OF $1.

This routine is the same as DALI except that only the maximal left ideals are evaluated.

DMRI GENERATE SET $2 OF ALL MAXIMAL RIGHT IDEALS OF $1.

This routine is the same as DARI except that only the maximal right ideals are evaluated.

DMTI GENERATE SET $2 OF ALL MAXIMAL TWO-SIDED IDEALS OF $1.

This routine is the same as DATI except that only the maximal two-sided ideals are evaluated.

DMNS GENERATE SET $2 OF ALL MAXIMAL NORMAL SUBGROUPS OF $1.

This routine is the same as DANS except that only the maximal normal subgroups are evaluated.

EGSS GENERATE SUBSEMIGROUP $3 FROM ELEMENT $2 OF $1.

This routine sets up the subsemigroup generated in a semigroup by one element as a subset of that semigroup.

EGLI GENERATE LEFT IDEAL $3 FROM ELEMENT $2 OF $1.

This routine sets up the left ideal generated by a given element of a given semigroup as a subset of that semigroup.

EGRI GENERATE RIGHT IDEAL $3 FROM ELEMENT $2 OF $1.

This routine sets up the right ideal generated by a given element of a given semigroup as a subset of that semigroup.

EGTI GENERATE TWO-SIDED IDEAL $3 FROM ELEMENT $2 OF $1.

This routine sets up the two-sided ideal generated by a given element of a given semigroup as a subset of that semigroup.

ABTE IS $1 ABELIAN, ANSWER $2.

A test is made to see whether the group (or semigroup) specified by the first parameter is Abelian. The second parameter is set to the logical value of the answer (1-ves. 0-no).

GRTE IS $1 A GROUP, ANSWER $2.

A test is made to see whether the semigroup specified by the first parameter is in fact a group. The second parameter is set to the logical value of the answer.

SGTE IS $1 A SEMIGROUP, ANSWER $2.

A test is made to see whether the Cayley table specified by the first parameter is in fact associative. The second parameter is set to the logical value of the answer.

GRTI IS $1 A GROUP, ANSWER $2, INVERSE TABLE $3.

This routine is the same as GRTE, except that if the answer is yes (i. e., the semigroup is a group), a map is generated, giving for each element its inverse.

SGIT ARE $1 AND $2 ISOMORPHIC, ANSWER $3, ISOMORPHISM $4.

The Cayley tables given by the first and second parameters are chacked for isomorphism. An isomorphism may involve renaming of the elements of one of the semigroups. The third parameter is set to the logical value of the answer. The fourth parameter is set to the map, from one group to the other, which constitutes the isomorphism (if found).

U2ST UNION $3 OF SUBSETS $1 AND $2.

The union of the two subsets given by the first two parameters is set up as a subset. No reference to the Cayley table of the semigroup of which they are subsets is required (for this routine and the next two, I2ST and C2ST).

I2ST INTERSECTION $3 OF SUBSETS $1 AND $2.

The intersection of the two subsets given by the first two parameters is set up as a subset.

CPST COMPLEMENT $2 OF SUBSET $1.

The complement of the subset mentioned in the first parameter is set up as a subset.

HMTE IS MAP $3 FROM $1 TO $2 A HOMOMORPHISM, ANSWER $4.

This routine checks whether the map given by the third parameter, with domain and range given respectively by the first two parameters, is a homomorphism. The fourth parameter is set to the logical value of the result.

MMTE IS MAP $3 FROM $1 TO $2 A MONOMORPHISM, ANSWER $4.

This routine checks whether the map given by the third parameter, with domain and range given respectively by the first two parameters, is a monomorphism (i. e., an isomorphism into). The fourth parameter is set to the logical value of the result.

EMTE IS MAP $3 FROM $1 TO $2 AN EPIMORPHISM, ANSWER $4.

This routine checks whether the map given by the third parameter, with domain and range given respectively by the first two parameters, is an epimorphism (i. e., a homomorphism onto). The fourth parameter is set to the logical value of the result.

IMTE IS MAP $3 FROM $1 TO $2 AN ISOMORPHISM, ANSWER $4.

This routine checks whether the map given by the third parameter, with domain and range given respectively by the first two parameters, is an isomorphism. The fourth parameter is set to the logical value of the result.

CTST CAYLEY TABLE $3 OF SUBSET $2 OF SEMIGROUP $1.

The subset given in the second parameter is expanded into a full Cayley table. (Note in review: Subsets are generally expressed by a table which has length equal to the order of the original semigroup. This may be greater than the size of the Cayley table of the subset; besides, the Cayley table format may be needed specifically, e. g., as input to other routines. On the other hand, note that converting to Cayley table format means that the structure of the subset within the original semigroup is disregarded by any reference to the new Cayley table.)

FGCT CAYLEY TABLE $3 OF FACTOR GROUP $1/$2.

The second parameter is assumed to refer to a subset of the Cayley table given by the first parameter; and this subset should in fact be a normal subgroup. The Cayley table of the factor group is set up and named according to the third parameter.

CTXB CAYLEY TABLE $3 IS IMAGE OF $1 UNDER PERMUTATION $2.

This routine is used principally to provide debugging input for the isomorphism test routines. The elements of the Cayley table given in the first parameter are renamed, according to the map given in the second parameter. A new Cayley table is produced, with the third parameter as its name.

DUCM DUMP MEMORY.

A diagnostic dump of all constants and tables is provided. Note: For Cayley tables, the numbers given in a memory dump will be one greater than the numbers typed in or printed out. This is because ALGBRA handles elements of a Cayley table in a manner compatible with FORTRAN (MADTRAN) conventions. The same will be true for maps.

EGNS GENERATE NORMAL SUBGROUP $3 FROM ELEMENT $2 OF $1.

This routine sets up the normal subgroup generated by a given element of a given group as a subset of that group.

SGSS GENERATE SUBSEMIGROUP $3 FROM SUBSET $2 OF $1.

This routine sets up the subsemigroup generated by a given subset of a given semigroup as a subset of that semigroup. The second and third parameters may be the same; i. e., the new subset may replace the old in the computer.

SGLI GENERATE LEFT IDEAL $3 FROM SUBSET $2 OF $1.

This routine sets up the left ideal generated by a given subset of a given semigroup as a subset of that semigroup.

SGRI GENERATE RIGHT IDEAL $3 FROM SUBSET $2 OF $1.

This routine sets up the right ideal generated by a given subset of a given semigroup as a subset of that semigroup.

SGTI GENERATE TWO-SIDED IDEAL $3 FROM SUBSET $2 OF $1.

This routine sets up the two-sided ideal generated by a given subset of a given semigroup as a subset of that semigroup.

SGNS GENERATE NORMAL SUBGROUP $3 FROM SUBSET $2 OF $1.

This routine sets up the normal subgroup generated by a given subset of a given semigroup as a subset of that semigroup.

CTAZ ADD ZERO TO SEMIGROUP $1 GIVING $2.

This routine performs the operation of adding a zero element to a given semigroup.

CTAU ADD UNIT TO SEMIGROUP $1 GIVING $2.

This routine performs the operation of adding a unit element to a given semigroup.

LCMA CONSTRUCT LEFT COSET MAP $3 OF SUBSET $2 OF $1.

The natural map, sending each element of the group given by the first parameter into the left coset (modulo the subgroup given by the second parameter) to which it belongs, is constructed and given a name by the third parameter.

RCMA CONSTRUCT RIGHT COSET MAP $3 OF SUBSET $2 OF $1.

The natural map, sending each element of the group given by the first parameter into the right coset (modulo the subgroup given by the second parameter) to which it belongs, is constructed and given a name by the third parameter.

PHRASE CLEAR MEMORY.

PHRASE is an entry point in the routine which analyzes the calling phrases. The effect of typing the above phrase is to "start over." Thus it is not necessary to reload ALGBRA.

ZETE DOES $1 HAVE A ZERO, ANSWER $2.

The semigroup given by the first parameter is checked. If it contains a zero (an element z such that zs = sz = z for all s) then the second parameter is set to 1; otherwise, 0.

IDNU HOW MANY IDEMPOTENTS IN SEMIGROUP $1, ANSWER $2.

The number of idempotents (elements e with ee = e) in the semigroup given by the first parameter is placed in the second parameter. The idempotent count will always be at least 1, and for a group it will always be exactly 1.

EONU SEMIGROUP $1, UNIT $2, IS FOUND TO HAVE
$4 ELEMENTS OF ORDER $3.

The semigroup (actually, in ALGBRA 1.4, this must actually be a group) with unit element specified is searched for all elements of the given order, and the number of these elements is placed in the fourth parameter. (Note: An element of a semigroup has, in general, two orders. Namely, if i is the first power of the given element which is equal to a preceding power j, then the first order is i-j, and the second order is j. for groups, the second order is always 1, and the first order is the order. A future routine will be written which will count elements by a specification of first and/or second orders.)

SHGT IS SEMIGROUP $1 WITH UNIT $3 A GROUP, ANSWER $2.

The semigroup with unit specified by the first and third parameters is checked to see whether it is a group. The second parameter is set to the logical value of the result. If it is already known that a semigroup has a unit, this routine is faster than the group test GRTE.

SUGU   IS SEMIGROUP $1 WITH UNIT $3 A GROUP,
        ANSWER $2. INVERSE TABLE $4

This routine is the same as SUGT, except that if the semigroup with unit is a group, the inverse table is left in the subset specified by the fourth parameter.

GRTU   IS $1 A GROUP, ANSWER $2, UNIT $3.

This routine is the same as GRTE, except that the unit is returned to the third parameter (if a unit exists, whether the semigroup is a group or not).

GRUT   IS $1 A GROUP, ANSWER $2, UNIT $3, INVERSE TABLE $4.

This routine is the same as GRTT, except that both the unit and the inverse table are returned. The unit is returned if a unit exists, whether the semigroup is a group or not.

UNTE   DOES $1 HAVE A UNIT, ANSWER $2.

The semigroup given by the first parameter is checked. If it contains a unit (an element e such that es = se = s for all s) then the second parameter is set to 1; otherwise, 0.

UNTU   DOES $1 HAVE A UNIT, ANSWER $2, UNIT $3.

This routine is the same as UNTE, except that if the semigroup has a unit, it is returned to the location given by the third parameter.

DUSS   PRINT SET OF SUBSETS $1.

The set of subsets given by the first parameter (such as a set of subsemigroups returned by DASS) is displayed on the console in the same format as that used by the routine (DUST) which displays a single subset.

ADSM   ADD $3 AS MAP NUMBER $2 IN SET $1.

This routine allows a map to be added to a set of maps. The set of maps may thereby expand, or it may remain the same size, depending on the value of the second parameter and the size of the set of maps.

MESM   MAP $3 IS NUMBER $2 IN SET $1.

This routine allows a map to be extracted from a set of maps. It may also be used to extract a subset from a set of subsets.

DUSM   SYSTEM DUMP.

A dump is given of all tables within the system in a form which does not depend on their being meaningful. Thus a dump may be taken under conditions that would cause the other dump routine (DUMP MEMORY) to "blow up."

* * * * * * * * * * * *

BIBLIOGRAPHY

1. W. D. Maurer, *Computer Experiments In Finite Algebra*. Brown University, January 1965