Zero divisors and invariant subspaces

(extended abstract)
by

Lajos Rónyai

Department of Computer and Information Science
University of Oregon

Computer and Automation Institute
Hungarian Academy of Sciences

1. Introduction

In this paper we continue the study of algorithmic problems related to associative algebras we initiated in Friedl - Rónyai [2]. The main result of this paper is an algorithm to find zero divisors in associative algebras over finite fields. The idea of this algorithm comes from an almost constructive proof of Weddernburn's theorem on finite fields given in Herstein [3].

The above result is applied to the problem of finding common invariant subspaces for a set of matrices over a finite field. Using this method one can give a polynomial time algorithm to find certain minimal normal subgroups in permutation groups. The corresponding questions over fields of characteristic zero remain open.

For the basic notions of the algebra used, the reader is referred to Friedl - Rónyai [2]. By an f-algorithm we mean an algorithm which uses an oracle to factor polynomials over finite fields. The complexity of an oracle call will be the length of the representation of the field plus the length of the dense representation of the polynomial in question.

2. Zero divisors in full matrix algebras

In this section we develop our basic reduction methods for finding zero divisors in full matrix algebras over finite fields. Let $M_n(Z)$ denote the algebra of n by n matrices over the finite field $Z=GF(q)$. Let a stand for an element of $M_n(Z)$ which is not in Z and for which $F=Z(a)$ is a field of dimension l over Z (i.e. the minimal polynomial of a is irreducible of degree l over Z).

Lemma 2.1. There exists a $c\in M_n(Z)$ such that
i) $c^{-1}ac=a^q$
ii) if $Alg(a,c)$ denotes the Z subalgebra generated by a and c then $Alg(a,c)$ is a non commutative Z algebra

iii) $Alg(a,c)=F+cF+c^2F+...+c^nF+...$ where + stands for (not necessarily direct) sum of Z subspaces.

Proof. F is a simple subalgebra of $M_n(Z)$ containing Z and the automorphism of F sending a to a^q leaves Z fixed elementwise, so by Noether - Skolem's theorem (cf. Herstein [3]) this automorphism must be inner, showing the validity of i). Now for the rest of the proof, let c be an arbitrary element satisfying i). From the fact, that a is not in Z, it follows that $a\neq a^q$ hence $Alg(a,c)$ is not commutative. To prove iii) it is enough to observe that $ac=ca^q$.

Lemma 2.2. Let a,c,l,F be as above. If c^l is not in Z then $Alg(a,c)\neq M_n(Z)$. If $c^l\in Z$ then

(1) $\quad Alg(a,c)=F+cF+...+c^{l-1}F$

and if $Alg(a,c)=M_n(Z)$ then $l=n$, the above sum is a direct sum and if f is the minimal polynomial of c over Z, then $deg(f)=n$.

Proof. A straightforward calculation shows that $c^{-i}ac^i=a^{q^i}$ for each nonnegative integer i. This implies that $c^la=ac^l$, i.e. c^l is in the center of $Alg(a,c)$. Now, observing that the center of $M_n(Z)$ is Z, the first statement follows. If c^l is in Z then any power of c is a Z linear combination of the elements 1, c, $c^2,...,c^{l-1}$ so (1) follows from lemma 2.1 and from the fact that F is a Z subspace. On the other hand, l is the degree of the mimimal polynomial of a over Z, hence $l\leq n$. From (1) it follows that $dim_Z Alg(a,c)\leq l^2$ and equality is attained if and only if the sum of subspaces is a direct sum. If $Alg(a,c)=M_n(Z)$, then these facts imply that $n=l$, 1, c, $c^2,...,c^{n-1}$ are linearly independent over Z and the statements follow.

Next we shall focus on the case $Alg(a,c)=M_n(Z)$. We have seen that c is a root of a polynomial of form $x^n-\lambda$ where $\lambda\in Z$. Using this element λ we can explicitly construct a pair of zero divisors in $M_n(Z)$. As we shall see, it requires to find elements in F with a given norm. For an element d of F $norm(d)$ is defined as the product $norm(d)=dd^qd^{q^2}...d^{q^{n-1}}$ (i.e. it is really the norm relative to Z). The following lemma is a very important part of the argument in a proof of Wedderburn's theorem about finite division algebras (see Herstein [3], pp. 71-72).

Lemma 2.3. Let d be an element of F such that $norm(d)=\frac{1}{\lambda}$. Then the element $1-cd\in Alg(a,c)$ is a zero divisor in $Alg(a,c)=M_n(Z)$.

Proof. Let us define the element $z\in Alg(a,c)$ as

$$z=1+cd+c^2dd^q+...+c^{n-1}dd^q...d^{q^{n-2}}.$$

A straightforward calculation shows that $z(1-cd)=o$. On the other hand, the fact that (1) is a direct sum implies that neither z nor $1-cd$ can be 0, proving the claim.

It turns out that we need more information about the element c to be able to solve the above norm equation in F.

Lemma 2.4. Suppose that $Alg(a,c)=M_n(Z)$ as before and suppose further that $x^n-\lambda$, the minimal polynomial of c over Z is irreducible in $Z[x]$. Then the

polynomial $g(x)=x^n-\frac{1}{\lambda}$ is also irreducible in $Z[x]$. Moreover the polynomial $g(x)$ splits into linear factors in the field F and if n is odd and $d\in F$ such that $g(d)=0$ then $norm(d)=\frac{1}{\lambda}$.

Proof. The irreducibility of $g(x)$ means that $Z(c)$ is a field of degree n over Z. On the other hand, $Z(c)=Z(\frac{1}{c})$, so the minimal polynomial of $\frac{1}{c}$ over Z has degree n, therefore it must be $g(x)$. Observing that $dim_Z F=n$, we see that F is isomorphic to $Z(\frac{1}{c})$ and the second statement follows. As for the last statement, let d be an arbitrary root of g. The irreducibility of g implies that its constant term can be written as $(-1)^n norm(d)=-norm(d)=-\frac{1}{\lambda}$ and this gives the result required.

We shall need a fact on certain subalgebras of $M_n(Z)$. The proof is a routine calculation and is therefore left to the reader as an exercise.

Lemma 2.5. If e is an idempotent element of $M_n(Z)$ of rank l then the algebra $eM_n(Z)e$ is isomorphic to $M_l(Z)$.

3. An algorithm to solve certain norm equations

In this section we outline an effective f-algorithm to solve norm eqations arising from lemma 2.3. More precisely, suppose that F is an n dimensianal extension field of $Z=GF(q)$, $q=p^r$, p is a prime, $f(x)=x^n-\beta\in Z[x]$ an irreducible polynomial over Z. We want to find an element $d\in F$ such that $norm(d)=\beta$. For our purposes, it is enough to deal with the case where n is either odd or $n=2$.

In the first case, as we have shown in lemma 2.4. , it is enough to find a root of f in F. This task can be solved by factoring f over F.

If $n=2$ then we shall distinguish two cases:

Case 1: $-\beta$ is a (quadratic) nonresidue in Z. Then if d is a root of the polynomial $g(x)=x^2+\beta$ in F then the other root from F must be d^q and calculating again the constant term of $g(x)$, we obtain that $d^{q+1}=norm(d)=\beta$.

Case 2: $-\beta$ is a residue in Z. Let e be an element of Z for which $e^2=-\beta$. Suppose that we can efficiently find an element $b\in F$ such that $norm(b)=b^{q+1}=-1$. Then letting $d=be$ we obtain that $norm(d)=e^2b^{q+1}=\beta$ as we wanted. Now we describe how to solve the norm equation $norm(b)=-1$. First we notice that in this case -1 is a nonresidue in Z because β is a nonresidue in Z, and it is the only nonresidue in Z having (multiplicative) order a power of two. This means that it is enough to find an element $b\in F$ such that the order of b is a power of two and $norm(b)=b^{q+1}$ is a nonresidue in Z. To this end, we define a sequence of elements of F: let $z_1=-1$. Suppose that z_i is defined. Then let z_{i+1} be an element of F such that $z_{i+1}^2=z_i$, provided that such element exists. Let z_k be the last element of this sequence. It is immediate that z_k is a nonresidue in F and it generates a

(multiplicative) group of order 2^k. The latter fact implies that $k \leq 2log_2 q$, so a z_k can be found by solving at most $2log_2 q$ quadratic equations in F. To prove that the choice $b=z_k$ is good, we remark that if b^{q+1} were a residue in Z then it would imply that $b^{\frac{(q+1)(q-1)}{2}}=1$, but this is impossible because b is a nonresidue in F.

What we obtained is the following

Lemma 3.1. The norm equation described at the beginning of this section can be solved by an f-algorithm running in time polynomial in n, r and $logp$.

4. Finding zero divisors

4.1. A reduction procedure

First we describe an auxiliary procedure to locate zero divisors in full matrix algebras over finite fields. The procedure CUT() has a single input parameter A which is required to be a noncommutative full matrix algebra over a finite field. The algebra A can be given by structure constants over its prime field $GF(p)$. It returns either a pair of zero divisors or a proper noncommutative subalgebra of A (which is generated by two elements over the center of A). It will be an f-algorithm running in time polynomial in $log(p)$ and m, the dimension of A over $GF(p)$.

procedure CUT(A)

Step 1. Find Z the center of A.
(* Here we may suppose that $A=M_n(Z)$ where $Z=GF(q)$, $q=p^r$, $m=rn^2$ and $n \geq 2$ *)

Step 2. Pick an arbitrary noncentral element b of A and compute and factor its minimal polynomial f over Z. If this is reducible over Z and $f=gh$ a proper factorization then return $g(b)$ and $h(b)$ as a pair of zero divisors.

Step 3. (* Here we know that f is irreducible, so $Z(b)$ is a field *)
Find an element $a \in Z(b)Z$ such that if $F=Z(a)$ then $l=dim_Z F$ is either 2 or it is an odd number.

Step 4. Find a nonzero element c of A for which $ac=ca^q$ (by solving a system of linear equations). Compute and factor the minimal polynomial of c over Z. If it is reducible then return zero divisors as in Step 2.

Step 5. (* Here we know that c is an invertible element of A and that $c^{-1}ac=a^q$ *)
Form $Alg(a,c)$ the Z algebra generated by a and c. If $Alg(a,c) \neq A$ then return $Alg(a,c)$ as a proper noncommutative subalgebra of A.

Step 6. (* At this point $n=l$, $Alg(a,c)=A=M_n(Z)$ and n is either 2 or it is odd, the minimal polynomial of c over Z is $f(x)=x^n-\lambda$ for some $\lambda \in Z$ and f is

irreducible over Z *)
Find a solution d of the norm equation $norm(x)=\frac{1}{\lambda}$ in F using the algorithm described in section 3 and return the pair of zero divisors $1-cd$ and $1+cd+c^2dd^q+...+c^{n-1}dd^q...d^{q^{n-2}}$.

end procedure

Lemma 4.1. Procedure CUT() is correct and it runs in time polynomial in m and $log(p)$ as an f-algorithm.
Proof. (Outline) If we terminate at Steps 2 or 4 then we have indeed found a pair of zero divisors (see [2] prop. 6.5). Step 3 can be done by solving a system of linear equations describing an appropriate subfield of F. If we terminate at Step 5 then by lemma 2.1 i) we have a proper noncommutative subalgebra of A. If termination occurs at Step 6 then lemmas 2.3, 2.4 and 3.1 guarantee the desired result. The timing follows from lemma 3.1.

4.2 The main algorithm

We are in a position to describe the key method of this paper. We have an associative algebra A over the finite field $Z=GF(q), q=p^r, p$ prime and $dim_Z A=m$. Our objective is to find a pair of zero divisors, i.e. nonzero elements $x,y\in A$ such that $xy=0$ if there are any.

We outline the major steps of our procedure ZERODIV(). Its input is an algebra over a finite prime field.

procedure ZERODIV(A)

Step 1. Compute $Rad(A)$ using the algorithm of [2]. If $Rad(A)\neq(0)$ then pick an arbitrary nonzero element $x\in Rad(A)$. As x is nilpotent, an appropriate power of it will suffice as y and terminate.

Step 2. (* A is semisimple *)
Determine the Wedderburn Artin decomposition of A, by using the decomposition algorithm of [2] . If A is not simple, say $A=I+J$ where I,J are proper ideals of A and the sum is a direct sum then x and y can be arbitrary nonzero elements of I and J respectively; terminate.

Step 3. (* A is simple *)
Check whether A is commutative. In case of affirmative answer terminate concluding that A is a field (i.e. it does not contain zero divisors).

Step 4. (* A is a complete matrix ring over some extension of Z, say $A=M_n(L)$ where $n\geq 2$ and L is a finite field containing Z *)
Call CUT(A). If it returns a pair of zero divisors from A then terminate. Otherwise it returns a proper subalgebra of form $Alg(a,c)$ of A. In this case let $A:=Alg(a,c)$ and go back to Step 1.

Now we prove the correctnes of ZERODIV(). Let d be the dimension of A over its prime field (i.e. $d=rm$). First we observe that ZERODIV() is essentially a loop and each iteration decreases the dimension of the actual A, so the number of iterations is not more than d. If CUT() returns a smaller algebra of form $Alg(a,c)$ then it is not commutative, so by Wedderburn's theorem on finite division rings, it must contain zero divisors, therefore in this case the algorithm can not terminate at Step 3. The validity of the annotation (i.e. the comments made) follows from the Wedderburn Artin structure theorem, in particular, the precondition of CUT() is fulfilled when it is called. Now the correctness follows from lemma 4.1.

We turn to the question of timing. We shall show that ZERODIV() as an f-algorithm runs in time polynomial in $log(p)$ and d. To see this, it is enough to see that each Step runs in time polynomial in $log(p)$ and d as an f-algorithm. For Steps 1 and 2 it follws fom theorems 5.7 and 7.8 of [2]. Step 3 can be done by solving a system of linear equations of size polynomial in $log(p)$ and d. The timing of Step 4 is established in lemma 4.1.

We can summarize this as follows:

Theorem 4.1. Let A be a d dimensional associative algebra over a prime field $GF(p)$. Then there exists an f-algorithm running in time polynomial in d and $log(p)$ to find zero divisors in A (if there are any).

5. Applications

In this section we shall apply our algorithm ZERODIV() to derive algorithms for some more interesting questions. First we give an algorithm to construct explicit isomorphism between matrix algebras. Next we describe a method to find common invariant subspaces for a set of matrices over a finite field.

5.1. Explicit isomorphisms of matrix algebras

From theorems 5.7 and 7.8 of [2] it follows that there exist an efficient f-algorithm to decide whether a given finite algebra A is isomorphic to a full matrix algebra and if the answer is yes, say $A=M_n(Z)$ then we can also find n and Z. Our aim here is to establish the above isomorphism explicitly: we want to construct a mapping from A to the algebra of n by n matrices over Z. To do this, it is enough to construct an n dimensional vectorspace V over Z on which A acts faithfully as an algebra of linear transformations. Indeed, then comparing dimensions immediately gives that the image of A (which is isomorphic to A because A is a simple algebra) must be the algebra of *all* linear transformations of V, so if we pick an arbitrary basis of V, then we obtain a representation of A by n by n matrices over Z. To construct such a vectorspace and action it is enough to find an idempotent $e\in A$ which has rank 1 in $M_n(Z)$ (in the light of lemma 2.5, this rank is independent from the actual isomorphism). Indeed, it is well-known that in this case if $V=M_n(Z)e$, then $dim_Z V=n$ (this is essentially the set of matrices with all entries zero except possibly the entries in the first column) and $M_n(Z)$ acts nontrivially therefore faithfully on V via multiplication from the left.

Now we outline our algorithm IDEMPOTENT() which has one input parameter A and A is expected to be isomorphic to a finite full matrix algebra. It

returns an idempotent e of rank one (or, equivalently, an idempotent for which eAe is a field).

Procedure IDEMPOTENT(A)

Step 1. Call ZERODIV(A). If A does not contain zero divisors then return the identity element of A.

Step 2. (* Here we have a zero divisor $x \in A$ *)
Find the right identity element e of the left ideal Ax by solving a system of linear equations.

Step 3. (* At this point we have an idempotent e which must be singular, for it is a zero divisor *) Return IDEMPOTENT(eAe).

end procedure

If initially A is isomorphic to $M_n(Z)$ and we are at Step 3 then lemma 2.5 shows that $eAe = M_l(Z)$ for some $l<n$ therefore the precondition of IDEMPOTENT() is satisfied and the number of calls can not be more than $n-1$. Suppose that we execute Step 2 $k \geq 1$ times, producing idempotents $e_1, e_2, \cdots, e_k$; then using the fact that $e_j e_i = e_i e_j = e_j$ if $i<j$, we obtain that at the last call of IDEMPOTENT() the parameter passed is $B = e_k M_n(Z) e_k$. The fact that this was the last call, implies that B is a field, and by lemma 2.5 e_k must be rank one in $M_n(Z)$. On the other hand, e_k is the identity element of B, so the algorithm works correctly in this case. If $k=0$ then $n=1$ and the correctness is obvious.

Now we deal with the running time. Let $Z = GF(p^r)$. It is clear that we always work in an algebra of dimension not more than rn^2 over $GF(p)$, therefore the calls of ZERODIV() (as an f-algorithm) take $poly(n,r,log(p))$ units of time. The remaining work is linear algebra which can also be done in time polynomial in n, r, $log(p)$. Now we can state the following

Theorem 5.1. Let A be an associative algebra over a finite prime field $GF(p)$ isomorphic to $M_n(GF(p^r))$ for some n and r. There exist an f-algorithm which runs in time polynomial in n, r, and $log(p)$ (i.e. polynomial in the input size) to construct an explicit isomorphism between A and $M_n(GF(p^r))$.

Proof. After having the procedure IDEMPOTENT() at hand, the remaining steps of constructing an isomorphism can clearly be done in polynomial time (without calling a factoring oracle).

Once we have an explicit isomorphism, we can decompose A into a direct sum of minimal left ideals. If e_{ii} denotes the matrix in $M_n(Z)$ which contains 1 in the i-th position of the i-th row and all other entries are zero, then
$M_n(Z) = M_n(Z)e_{11} + M_n(Z)e_{22} + \ldots + M_n(Z)e_{nn}$ is a decomposition into minimal left ideals. The images of the elements e_{ii} in A will give the decomposition desired. We have the following

Corollary 5.1. Let p, n, r, A be as in theorem 5.1. Then there exists an f-algorithm running in time polynomial in n, r and $log(p)$ to compute a decomposition of A as a direct sum of minimal left ideals.

We can generalize this one step further. If A is a semisimple algebra over Z then first we can decompose it into a direct sum of its minimal ideals (which are full matrix algebras) using the algorithm given in [2], next we decompose these ideals into a direct sum of minimal left ideals using the above algorithm. Putting these together, we obtain a decomposition of A into a direct sum of minimal left ideals.

Corollary 5.2. Let A be a semisimple algebra of dimension m over the finite field $Z=GF(p^r)$. There exists an f-algorithm running in time polynomial in m, r and $log(p)$ to compute a decomposition of A into a direct sum of minimal left ideals.

5.2. Common invariant subspaces

Consider the following problem. Given are matrices $X_1, X_2,...,X_k \in M_n(Z)$ and we consider their action on Z_n, the space of column vectors of length n with entries from Z. We want to decide whether they have a common invariant subspace, i.e. a proper Z-subspace $U \subseteq Z_n$ such that $X_iU \subseteq U$ for every $1 \leq i \leq k$. We shall give an f-algorithm which has time complexity polynomial in k, n, r and $log(p)$. Our algorithm will also produce such an invariant subspace if it exists.

The procedure INVARIANT() has one input parameter, which is a set of n by n matrices over a finite field Z. It outputs either a proper invariant subspace of Z_n or a message saying that there exists no such a subspace.

procedure INVARIANT(Φ)
(* $\Phi=\{ X_1, X_2, ...,X_k\}$ *)

Step 1 Compute (a basis of) A, the matrix algebra generated by Φ.
(* $U \leq V_n$ is an invariant subspace for Φ if and only if it is an invariant subspace for A. *)

Step 2 Compute $Rad(A)$. If $Rad(A) \neq (0)$ then return $U:=Rad(A)V_n$.

Step 3 (* At this point A is semisimple. *)
Compute a decomposition of A as a direct sum of minimal left ideals
$A=\rho_1+\rho_2+...+\rho_l$
Next select an arbitrary nonzero vector v from V_n and form the (A invariant) subspaces $\rho_1 v$, $\rho_2 v$,...$\rho_l v$ and let U be any of these which is not (0). If $U=V_n$ then there is no proper A-invariant subspace, otherwise return U.

end procedure

Now we prove the correctness of INVARIANT(). It is obvious that Φ and A have the same invariant subspaces. If ρ is a left ideal of A then any ρ invariant subspace is an A invariant subspace as well, therefore U is an A invariant

subspace upon termination. If we terminate at Step 2 then $U \neq (0)$ because A acts faithfully on V_n. On the other hand, $Rad(A)$ is a nilpotent algebra of matrices, so $U = V_n$ is impossible.

If we enter Step 2 then A is certainly semisimple, so it can be decomposed as a direct sum of minimal left ideals. It is known (see for example Herstein [3] pp. 97-98) that $\rho_j v$ is either (0) or it is a minimal A invariant subspace (or, with module theoretic terminology, it is an irreducible left A module) and not all of them can be (0). We conclude that U is a minimal A invariant subspace. In particular, if $U = V_n$ then V_n has no proper A invariant subspaces. The correctness is proved.

Step 1 can clearly be done in time polynomial in k, n, r and $log(p)$ and the dimension of A over Z is not more than n^2. Now using the algorithms of theorem 5.7 of [2] and corollary 5.2 we see that steps 2 and 3 can be done in time polynomial in n, r and $log(p)$ and we call a factoring oracle only at Step 3. We have the following:

Theorem 5.2. Let $\Phi = X_1,\ldots,X_k$ be a set of n by n matrices over the finite field $Z = GF(p^r)$. There exists an f-algorithm with time complexity polynomial in k, n, r and $log(p)$ to find a proper Φ invariant subspace in V_n (if there is any).

We remark that if A is semisimple then its action on V_n is completely reducible, i.e. V_n can be decomposed as a direct sum of minimal A invariant subspaces (c.f. Herstein [3] pp. 97-98). Using the last step of our algorithm INVARIANT(), one can show that the standard textbook decomposition process can be done in polynomial time as an f-algorithm. Indeed if we have a subspace U which is a direct sum of minimal A invariant subspaces $U_1, U_2, \ldots U_m$ we have already found, then we do Step 3 with a vector v which is not in U. A straightforward reasoning shows that at least one of the nonzero minimal A invariant subspaces $\rho_j v$ must intersect U trivially, thus giving a bigger direct sum.

Corollary 5.3. If Φ generates a semisimple subalgebra A of $M_n(Z)$ then V_n can be decomposed as a direct sum of minimal Φ invariant subspaces using an f-algorithm running in time polynomial in k, n, r and $log(p)$.

5.3. An application to permutation groups

Some computational problems in permutation groups can be reduced to the problem of finding common invariant subspaces over very small fields. The interested reader is referred to Babai, Kantor, Luks [1].

Here we shall consider the following situation. G is a permutation group on n letters $K < H$ normal subgroups of G and H/K is elementary abelian p group for some prime p. We may suppose that the above permutation groups are given by strong generating sets. Our aim is to find a minimal normal subgroup L of G such that $K < L \leq H$.

First we remark, that $V = H/K$ can be viewed as a vectorspace over $GF(p)$ of dimension $O(n log(n))$ and that G acts on this vectorspace (via conjugation) as a group of linear transformations. Our problem is equivalent with finding a minimal G invariant subspace in V. Obviously it is enough to find a minimal Φ invariant subspace where $\Phi = \{ g_1, g_2, \ldots, g_k \}$ is the strong generating set of G we

have. We can also compute a basis for V using the strong generating sets of H and K in time polynomial in n. The elements of Φ can be represented as matrices with respect to this basis and we can apply our algorithm INVARIANT() to find a minimal invariant subspace.

Corollary 5.4. Let $G \leq S_n$ and $K < H$, normal subgroups of G given by strong generating sets and H/K elementary abelian. There exists an algorithm running in time polynomial in n to find a minimal normal subgroup L of G such that $K < L \leq H$ hold.

Proof. It is enough to remark that $k \leq n^2$ and $p \leq n$, so we can use here the deterministic "exponential" factoring algorithm and the rest follows from theorem 5.2.

REFERENCES

[1] L. Babai, W. M. Kantor, E. M. Luks, *Computational complexity and the classification of finite simple groups,* Proc. 24th IEEE FOCS, Tucson, Arizona, 1983, 162-171.

[2] K. Friedl, L. Rónyai, *Polynomial time solutions of some problems in computational algebra,* Proc 17th ACM STOC, Providence, Rhode Island, 1985, 153-162.

[3] I. N. Herstein, *Noncommutative rings,* Math. Association of America, 1968.