COMPUTABLE ALGEBRA, GENERAL THEORY AND THEORY OF COMPUTABLE FIELDS[1]

BY

MICHAEL O. RABIN[2]

Introduction. Abstract algebra is the product of a transition from such concrete systems as the ring of integers or the field of rational numbers to the corresponding, axiomatically defined, abstract systems. In the course of this abstraction process various properties of the concrete system are lost. The field of rational numbers, for example, possesses a natural topology, an ordering, and furthermore the field operations, that is addition and multiplication of rationals, are effectively computable functions. None of these properties enters into the axiomatic definition of a field. Various attempts were made to reincorporate some of these features into the study of algebraic systems. In topological algebra we study groups endowed with a topology subject to the condition that the algebraic operations are continuous functions with respect to this topology. The basic definitions and the concepts studied are a blend of algebra and topology. Thus we are most interested in those homomorphisms which are also continuous mappings.

In this paper we follow a similar program and study a blend of algebra and theory of recursive functions. We consider *computable algebraic systems*; these are, roughly speaking, algebraic systems for which the algebraic operations are effectively computable functions (i.e. recursive functions; compare however the remarks in the concluding §2.5). Another concept of basic importance is that of a *computable homomorphism.*

Chapter I is devoted to the general theory of computable algebra. To avoid cumbersome statements and notations we confine our attention to the case of computable groups. We shall, however, indicate how the definitions and results extend to the case of computable rings. We start by proving general theorems about computability of factor groups and subgroups of a computable group. In analogy with the pattern of topological groups we introduce the computable structure into factor groups and subgroups in such a way that the natural homomorphisms and isomorphisms involved are computable. We define the notion of solvability with respect to a finite set of generators of the word problem of a finitely generated (but not necessarily finitely related) group, and prove (Theorem 4) that a group has a solvable word problem if and only if it is computable. This entails that if II and II′

Received by the editors December 16, 1957 and, in revised form, June 12, 1959.

(1) The results in this paper were presented in a talk before the Summer Institute of Symbolic Logic in July, 1957 at Cornell University.

(2) Now at the Hebrew University, Jerusalem.

are finite presentations defining isomorphic groups then the word problem of Π is solvable (in the usual sense) if and only if the word problem of Π' is solvable. Next we give a simple example of a finitely generated noncomputable group by constructing a group with an unsolvable word problem and then applying Theorem 4. We conclude with a purely algebraic application: A necessary condition for a finitely generated group to possess a faithful matrix representation is that it be computable.

In Chapter II we study more closely the case of computable fields. The main result is that if F is a computable field then its algebraic closure $\overline{F}$ is a computable field and possesses a computable structure such that the imbedding isomorphism of F into $\overline{F}$ is computable (Theorem 7). It turns out that the classical Steinitz construction of the algebraic closure, which proceeds by a step by step adjunction of roots of polynomials to the ground field F, can not be applied to prove Theorem 7. Showing that $\overline{F}$, the outcome of this construction, is computable, would require an algorithm for factorization of polynomials (see Definition 9) over each of the intermediate fields; such an algorithm does not always exist. We therefore use a different construction, related to the one in Bourbaki [2, pp. 89–92], which together with the ideas introduced in Chapter I enables us to show that $\overline{F}$ is indeed computable. Theorem 7 now leads to a simple necessary and sufficient condition for the existence of an algorithm for factorization of polynomials over a given field (Theorem 8). Using this condition we give a simple proof for a result due to van der Waerden concerning the existence of an algorithm for factorization of polynomials over a separable extension field.

The early studies related to the present paper were mainly concerned with the existence of algorithms for the solution of algebraic problems. Thus we have Kronecker's algorithm for the factorization of polynomials with rational coefficients; also his explicit construction of simple algebraic extensions of a field. Van der Waerden, in a pioneering paper [11], proved that there does not exist a splitting algorithm applicable to every "explicit" field. Since this paper was written before the creation of the theory of recursive functions, the term "explicit" is actually never defined in a precise way. Fröhlich and Shepherdson, using recursive functions, obtained a precise and stronger form of van der Waerden's example. Their paper [4] contains a wealth of results and examples concerning explicit fields. The notions of explicit field and computable field, though intimately related, arise from slightly different points of view about computable algebra; see the remarks which follow Definition 5. The term "computable group" was introduced by Rice [10] and, independently, by the present author in his dissertation (Princeton, 1956).

Chapter I. General theory of computable systems

1.1. Basic definitions. Throughout this paper the term computable function will be used as synonymous with partial recursive function. The domains

of definition of the computable functions appearing will be, as a rule, recursive sets. In special cases it may of course happen that a computable function is everywhere defined and hence is recursive. We shall start by recalling two definitions from the theory of recursive functions.

DEFINITION 1. A subset R of the set of positive integers I is called *recursive* if its characteristic function (i.e. the function f such that $f(n)=1$ if $n\in R$ and $f(n)=0$ otherwise) is a computable (recursive) function.

DEFINITION 2. A subset $R\subseteq I$ of the integers is *recursively enumerable* if there exists a computable function g from integers to integers such that g maps I onto R.

The sequence of function values $g(0)$, $g(1)$, $g(2)$, $\cdots$, is sometimes called an *effective enumeration* of the elements of R.

REMARKS. It is not hard to see that the range of values of a computable function g defined only on a recursive subset of I is also a recursively enumerable set. Every recursive set is also recursively enumerable. The reader can convince himself that $R\subseteq I$ is recursive if and only if both R and $I-R$ are recursively enumerable. This last fact is of basic importance in the theory of effective computation procedures.

DEFINITION 3. An *indexing* of a set S is a one to one mapping i: $S\rightarrow I$ such that $i(S)$ is a recursive subset of I. If $s\in S$ then $i(s)$ will be called the *index* of s. If $j\in i(S)$ is an integer, s_j will denote the element of S having j for index.

Note that a set S possesses indexings if and only if it is at most countable.

DEFINITION 4. An indexing i of a group G is *admissible* if the function m from $i(G)\times i(G)$ into $i(G)$ defined by the condition

$$g_j g_k = g_{m(j,k)} \qquad \textit{for all } g_j,\ g_k \in G, \tag{1.1}$$

is a computable function. (By Definition 3, g_j, g_k, and $g_{m(j,k)}$, are the elements of G having the indexes j, k, and $m(j, k)$.) The function m is the *multiplication function corresponding to the indexing* i.

DEFINITION 5. A group is *computable* if it possesses at least one *admissible* indexing.

At this point the question arises why we do not adopt the following as our concept of a computable group: A group which is a recursive subset of the integers and for which the group operation is a computable function from pairs of integers to integers. Every group computable according to Definition 5 is certainly isomorphic to such a group. Fröhlich and Shepherdson [4] use a generalized form of the above proposal as their definition of an "explicit" algebraic system. Again, explicit systems are identical, up to isomorphisms, with computable systems. We prefer, however, the approach by computable systems, and this for two reasons. Firstly, it enables us to consider all possible ways of introducing a computable structure into a given abstract algebraic system. This is important because not all these computable structures (i.e. admissible indexings) are equivalent. Thus in Definition 7 the fixed homomorphism ϕ may be computable with respect to one pair of indexings i_1, i_2,

and not computable with respect to another pair; again, the conditions of Theorems 1 and 3 may hold for an indexing i and not hold for some other admissible indexing of the same group. Secondly, separating between the algebraic system and its computable structure simplifies proofs. The argument breaks naturally into two steps, a purely algebraic step involving only the algebraic system, followed by a demonstration that certain functions and sets are recursive.

In the definitions of admissible indexings and computable groups no assumption concerning the effective computability of group-inverses is made. This is in fact unnecessary because of the following

LEMMA 1. *Let i be an admissible indexing of the computable group G, the function $in(j)$ from $i(G)$ into $i(G)$ defined by*

$$in(j) = i(g_j^{-1}), \qquad j \in i(G),$$

is computable.

Proof. Let k_0 be the index of the unit element of G. We have[3]

$$in(j) = (\mu t)[t \in i(G) \ \& \ m(t, j) = k_0],$$

where m is the multiplication function corresponding to i. The function $in(j)$ is therefore computable.

DEFINITION 6. Let G be a group, i an admissible indexing, and m the corresponding multiplication function. The set of integers $i(G)$ with the operation $m: i(G) \times i(G) \to i(G)$, which is then a group, will be denoted by (G, i) and called a *recursive realization* of G.

The mapping $i: G \to (G, i)$ is of course an isomorphism of G onto the recursive realization.

DEFINITION 7. Let G and F be computable groups, i_1 and i_2 fixed admissible indexings of G and F. A homomorphism $\phi: G \to F$ is called *computable with respect to i_1 and i_2* if the naturally induced corresponding homomorphism $\bar{\phi}: (G, i_1) \to (F, i_2)$, i.e. the homomorphism $\bar{\phi}$ which makes the diagram

$$\begin{array}{ccc} G & \xrightarrow{\phi} & F \\ i_1 \downarrow & & \downarrow i_2 \\ (G, i_1) & \xrightarrow{\bar{\phi}} & (F, i_2) \end{array}$$

commutative, is a computable function from integers to integers. A computable homomorphism ϕ is called *strongly computable with respect to i_1 and i_2* if $i_2(\phi(G))$ is a recursive subset of $i_2(F)$.

---

[3] (μt) reads: The least t such that . . . . Since m is a computable function and furthermore for every j there exists at least one $t \in i(G)$ such that $m(t, j) = k_0$, it is well known that $in(j)$ is indeed computable; see [7, pp. 279–280].

Concepts such as that of a computable isomorphism are of course special cases of Definition 7.

REMARK. Notice that if ϕ is computable with respect to i_1 and i_2 then $i_2(\phi(G))$, being the same as $\bar{\phi}(i_1(G))$, is always a recursively enumerable set of integers.

1.2. **Homomorphism and subgroup theorems.** We shall now prove two general theorems concerning the behavior of computable groups under passage to homomorphic images and to subgroups. These results will be very useful tools, later on, for showing that certain groups are computable.

THEOREM 1. *If i is an admissible indexing of the computable group G and H is a normal subgroup of G such that $i(H)$ is a recursive set of integers, then the factor group G/H possesses an admissible indexing i_1 such that the natural homomorphism $\phi: G \to G/H$ is computable with respect to i and i_1.*

Proof.[4] View G/H as the set of cosets gH where $g \in G$. It is enough to prove the existence of a computable function $f: i(G) \to I$ satisfying:

(1.2) *For all t, $k \in i(G)$, $f(t) = f(k)$ if and only if $g_t H = g_k H$,*

(1.3) *for all $g \in G$, $f(i(g)) \in i(gH)$,*

(1.4) *the set $f(i(G))$ is a recursive set of integers.*

Indeed, assuming the existence of such a function, define a mapping

$$i_1: G/H \to I$$

by the equation

$$i_1(gH) = f(i(g)), \qquad g \in G. \tag{1.5}$$

Conditions (1.2) and (1.4) imply that i_1 is a well defined indexing of the group G/H. If $j = i_1(gH) = f(i(g))$ then by (1.3) there exists an element $g' \in gH$ with $i(g') = j$. Let m and m_1 be the multiplication functions corresponding to i and i_1 (Definition 4). If $j = i_1(gH) = f(i(g))$ and $k = i_1(g_1 H) = f(i(g_1))$ then

$$m_1(j, k) = i_1(gg_1 H) = f(i(gg_1)). \tag{1.6}$$

By the previous remark the integer $m(j, k)$ is the i index of an element in G belonging to the left coset of H to which gg_1 belongs; hence, by (1.6) and condition (1.2),

$$m_1(j, k) = f(m(j, k)).$$

Since f and m are computable functions so is m_1. The indexing i_1 is thus admissible and G/H is a computable group.

The function $f: i(G) \to i_1(G/H)$ is, because of (1.5), the homomorphism from (G, i) onto $(G/H, i_1)$ corresponding to the natural homomorphism $\phi: G \to G/H$. This function is, by assumption, computable. Thus the two statements of the theorem are proven.

[4] The author thanks the referee for pointing out two shortcuts in the original proof.

All that remains is to construct a computable function f satisfying (1.2), (1.3), and (1.4). Let $E(j, k)$ be the relation holding between integers j and k if and only if $g_jH = g_kH$. This amounts to

$$E(j, k) \equiv [j, k \in i(G) \,\&\, m(in(j), k) \in i(H)].$$

The function m and the set $i(G)$ are recursive by the definition of an admissible indexing; the function $in(j)$ is computable (Lemma 1); the set $i(H)$ is recursive by assumption. The relation $E(j, k)$ is therefore recursive(5). Define a function $f: i(G) \to I$ by the condition

$$f(i(g)) = \min_{g' \in gH} i(g').$$

If $j \in i(G)$ then $f(j) = (\mu k)E(j, k)$(3); f is thus a computable function. Conditions (1.2) and (1.3) obviously hold. Now $j \in f(i(G))$ if and only if $j \in i(G)$ and $f(j) = j$. Since f is computable this implies that $f(i(G))$ is a recursive set; i.e. f satisfies (1.4).

REMARK. The converse of Theorem 1 also holds: *Let i be an admissible indexing of the computable group G and let H be a normal subgroup of G. If G/H is computable and possesses an admissible indexing i_1 such that the natural homomorphism $\phi: G \to G/H$ is computable with respect to i and i_1, then $i(H)$ is a recursive set of integers.* Indeed, let $\bar{\phi}: (G, i) \to (G/H, i_1)$ be the homomorphism naturally induced by ϕ, then $\bar{\phi}$ is a computable function by assumption. Let k_0 be the i_1 index of the unit element of G/H then $j \in i(H)$ if and only if $\bar{\phi}(j) = k_0$ so that the test for membership in $i(H)$ is effective.

As we shall see in §1.3, the condition that the natural homomorphism is computable cannot be omitted from the above assumptions. We have, however, the following

THEOREM 2. *Let G be a finitely generated computable group and let i be an admissible indexing of G. If H is a normal subgroup of G such that G/H is a computable group, then $i(H)$ is a recursive set of integers.*

Proof. The proof will consist of showing that the natural homomorphism, and in fact every homomorphism of a finitely generated computable group into a computable group, is computable. Let G_1 be a computable group and i_1 an admissible indexing of G_1; let $\psi: G \to G_1$ be any homomorphism, and $\bar{\psi}: (G, i) \to (G_1, i_1)$ be the homomorphism induced by ψ.

In order to avoid complicated notations let us assume that $i(G) = I$. As G is finitely generated there exists a fixed integer n such that every $g \in G$ is a product, with positive exponents, of the elements $g_1, g_2, \cdots, g_n$ (where g_j is the element satisfying $i(g_j) = j$). Denote $i_1(\psi(g_j))$ by k_j. By an effective enumeration of the indexes of all products $p = g_{j(1)}g_{j(2)} \cdots g_{j(r)}$, where $1 \leqq j(1), \cdots, j(r) \leqq n$, it is possible to find effectively for every $j \in i(G)$ a

(5) A relation $E(j, k)$ is called *recursive* if the function $e: I \times I \to I$ defined by $e(j, k) = 1$ if $E(j, k)$ holds, $e(j, k) = 0$ otherwise, is a recursive function.

product representation $j=j(1)\circ j(2)\circ\cdots\circ j(r)$ ($\circ$ is the operation of (G, i)). Now $\bar{\psi}(j)=k_{j(1)}\circ k_{j(2)}\circ\cdots\circ k_{j(r)}$ ($\circ$ now being the operation m_1 of (G_1, i_1)). Since m_1 is a computable function the right side of the last equation can be computed effectively so that $\bar{\psi}$ is a computable function. According to Definition 7 this means that ψ is computable with respect to i and i_1. The proof of the theorem is now completed by appealing to the converse of Theorem 1.

THEOREM 3. *If i is an admissible indexing of the computable group G and S is a subset of G such that $i(S)$ is a recursively enumerable set of integers, then $G(S)$, the subgroup of G generated by S, is a computable group.*

Proof. Let $J'=i(S)$, then J' is a recursively enumerable set of integers. The function $in(j)$ defined in Lemma 1 is computable; the set $J''=in(J')=i(S^{-1})$ (where $S^{-1}=\{g\,|\,g^{-1}\in S\}$) is therefore also recursively enumerable. It follows that $J=J'\cup J''=i(S\cup S^{-1})$ is recursively enumerable. This in turn implies that the set $N=\{\langle j_1,\cdots, j_n\rangle\,|\,j_r\in J,\ n=1, 2,\cdots\}$ of all n-tuples (for all n) with coordinates in J can be enumerated effectively. The mapping $p: N\to I$ defined by $p(\langle j_1,\cdots, j_n\rangle)=j_1\circ\cdots\circ j_n$ (where $\circ$ is the operation m of (G, i)) is computable; the set $p(N)$ is therefore recursively enumerable. The elements of $p(N)$ are precisely the indices of all products of elements in $S\cup S^{-1}$, i.e. $p(N)=i(G(S))$. Thus if $i(S)$ is recursively enumerable then $R=i(G(S))$ is also recursively enumerable. If R is finite then $G(S)$ is finite and hence, trivially, a computable group. If R is infinite there exists a computable one to one function $f: I\to R$ such that $f(I)=R$. Define an indexing i_1 of $G(S)$ by

$$i_1(g)=f^{-1}(i(g)),\qquad g\in G(S).$$

i_1 is an indexing because it is one to one and $i_1(G(S))=f^{-1}(R)=I$ is a recursive set. For m_1, the multiplication function corresponding to i_1, we have $m_1(j, k)=f^{-1}(m(f(j), f(k))$ so that m_1 is computable.

COROLLARY. *Every finitely generated subgroup of a computable group is a computable group.*

1.3. **Examples.** We shall construct examples of computable groups and homomorphisms between these for the purpose of illustrating certain situations which may arise. In particular we shall show that it is impossible to weaken the assumptions in the converse to Theorem 1 and still retain the conclusion.

Let G be the weak direct product of the groups G_j, $j=1, 2,\cdots$, where G_j is the cyclic group of order p_j (the jth prime) on the generator x_j. The mapping $i: G\to I$ defined by

$$i((x_{j(1)})^{r_1}\cdots(x_{j(n)})^{r_n})=(p_{j(1)})^{r_1}\cdots(p_{j(n)})^{r_n},\qquad 0\leqq r_k\leqq p_{j(k)},$$

is clearly an admissible indexing of G (all the $j(k)$ are pairwise distinct).

Let R be a set of integers which is recursively enumerable but not recur-

sive; $I-R$ is therefore not recursively enumerable. The subgroup H of G generated by the x_j, $j\in R$, is a direct factor of G. Thus $G=H\otimes D$ where D is the subgroup generated by the x_k, $k\in I-R$. The set of i indexes $i(\{x_j|j\in R\})$ is clearly recursively enumerable so that H is, by Theorem 3, a computable group.

We contend that D is not a computable group. The set of equations $x^{p_k}=1$ which possess a solution $x\neq 1$ in a fixed computable group F is recursively enumerable. To get an effective enumeration simply substitute, in some effective order, each nonunit group element in each equation and compute, listing at the same time every equation that was satisfied. On the other hand, an equation $x^{p_k}=1$ has a solution $x\neq 1$ in D if and only if $k\in I-R$. Since $I-R$ is not recursively enumerable, D is not a computable group.

To sum up: We have constructed a computable group G which has a direct product decomposition $G=H\otimes D$ such that H is a computable group and D is not a computable group.

EXAMPLE 1. The group G/D, being isomorphic to H, is computable. For any admissible indexing i of G, however, $i(D)$ is not a recursive set; for if it were, the group D would be computable by Theorem 3. Thus the condition, in the statement of the converse to Theorem 1, that the natural homomorphism $\phi: G\to G/H$ is computable, cannot be deleted.

EXAMPLE 2. It can be shown that D is the kernel of any homomorphism $\psi: G\to H$ (where ψ is onto), regardless of whether ψ is the natural homomorphism or not. If any such homomorphism ψ were computable with respect to some admissible indexings i of G and i_1 of H, then $i(D)$ would be a recursive set by the converse to Theorem 1; this contradicts the conclusion in Example 1. We thus have two computable groups G and H such that H is an homomorphic image of G but none of the homomorphisms of G onto H is computable.

EXAMPLE 3. Finally let us observe that, since $G=H\otimes D$, we have $D\approx G/H$. Thus the factor group of a computable group by a computable subgroup need not be a computable group.

1.4. **The word problem of groups.** In this section the general concepts and results treated thus far are applied to obtain some insight and information concerning the word problem for groups.

DEFINITION 8. A finitely generated group G is said to have a *solvable word problem with respect to a system of generators* $(g_1, \cdots, g_n)$ if the set of words $u(x)$ on the symbols $X=\{x_1, \cdots, x_n\}$ for which the equation $u(x)=1$ is satisfied in G upon substituting $g_1, \cdots, g_n$ for $x_1, \cdots, x_n$, is a recursive subset of the set of all words on X.

It is easy to extend the definition to infinitely generated groups G and infinite sequences $S=(g_1, g_2, \cdots)$ of generators. But it may happen that by this extended definition the word problem of G with respect to S is solvable, whereas the word problem of G with respect to $S'=(g_1', g_2', \cdots)$, where S'

is merely a permutation of S, is not solvable. To avoid such undesirable possibilities we keep the concept restricted to finitely generated groups and finite systems of generators.

LEMMA 2. *The free group F on n generators $(x_1, \cdots, x_n)$ is computable.*

Proof. The elements of F are the reduced words $w(x)$,

$$w(x) = (x_{g(1)})^{e(1)}(x_{g(2)})^{e(2)} \cdots (x_{g(k)})^{e(k)}, \qquad k \text{ arbitrary}, \tag{1.7}$$

where $1 \leqq g(j) \leqq n$ and $e(j)$ is a positive or negative integer. Define an indexing i of F by

$$i(w(x)) = (p_1)^{g(1)} \cdots (p_k)^{g(k)}(p_{k+1})^{h(1)} \cdots (p_{2k})^{h(k)},$$

where p_j is the jth prime and $h(j)$ is $2|e(j)|$ or $2|e(j)|+1$ according as to whether $e(j)>0$ or not. It is easily seen that $i(F)$ is a recursive set of integers so that i is indeed an indexing of F. This indexing is effective in the sense that given a word w the number $i(w)$ is computable and given an integer $j \in i(F)$ the expression (1.7) of the word $i^{-1}(j)$ can be effectively determined. Given two integers j, $r \in i(F)$, the word $i^{-1}(j)i^{-1}(r)$ can be effectively determined and transformed to its reduced form w. Then $i(w)$ can be computed. As $i(w) = m(j, r)$, where m is the multiplication function corresponding to i, this shows that m is a computable function. The indexing i is thus an admissible indexing of F.

THEOREM 4. *A finitely generated group G has a solvable word problem with respect to a system $S = (g_1, \cdots, g_n)$ of generators if and only if G is a computable group.*

Proof. Denote by F the free group on the generators $X = (x_1, \cdots, x_n)$; as a set, F consists of all reduced words on X. Let U be the set of all reduced words $u(x)$ which reduce to 1 in G upon substituting g_i for x_i, $i = 1, \cdots, n$. A word $w(x)$ reduces to 1 under this substitution if and only if its reduced form does; U is therefore a recursive set if and only if the word problem of G with respect to S is solvable. The mapping $x_i \to g_i$, $i = 1, \cdots, n$, induces a homomorphism $F \to G$; the kernel clearly is U and therefore $F/U \approx G$.

By Lemma 2 F is a computable group; let i be the effective admissible indexing of F defined there. Theorems 1 and 2 imply that F/U, and hence G, is a computable group if and only if $i(U)$ is a recursive set of integers; that is, since i is effective, if and only if U is a recursive set. But U is recursive if and only if the word problem of G with respect to S is recursively solvable.

COROLLARIES. Theorem 4 immediately implies that *if the word problem of a group is solvable with respect to one system of generators it is solvable with respect to any other system of generators.* It is therefore permissible to speak about the solvability or unsolvability of the word problem of a finitely generated group without referring to any particular system of generators.

Since every finitely generated subgroup of a computable group is computable (corollary of Theorem 3), *the word problem of every finitely generated subgroup of a group having a solvable word problem, is itself solvable.* These two results fill the gap in [9, p. 187].

1.5. **A noncomputable group.** In this section a finitely generated group with an unsolvable word problem is constructed; by Theorem 4 this group is not computable. The construction is similar, in some respects, to an unpublished example by W. Boone though it differs from it in purpose and method.

Consider a presentation having the four generators x, y, u, t, and having defining relations of the form $u^ixu^{-i}=t^iyt^{-i}$ where i runs through a fixed set U of positive integers; denote the group defined by this presentation by G_U.

LEMMA 3. *In the group G_U defined by the presentation*

$$(x, y, u, t: u^ixu^{-i} = t^iyt^{-i}, i \in U)$$

a relation of the form $u^ixu^{-j}=t^iyt^{-j}$ holds if and inly if $j\in U$.

Proof. Nothing has to be said about the if part. Let F_1 be the free group on the generators x and u, and let $H_1\subseteq F_1$ be the subgroup generated by the elements u^ixu^{-i}, $i\in U$; similarly, let F_2 be the free group on y and t, and $H_2\subseteq F_2$ be the subgroup generated by the elements t^iyt^{-i}, $i\in U$. The correspondence $u^ixu^{-i}\leftrightarrow t^iyt^{-i}$, $i\in U$, clearly induces an isomorphism $H_1\approx H_2$; the group G_U is therefore the free product of F_1 and F_2 with the amalgamation $H_1=H_2$,

$$G_U \approx (F_1 * F_2)_{H_1=H_2}.$$

It follows from the basic properties of the free product with amalgamated subgroups that in G_U an equation of the form $w(x, u)=w(y, t)$, where $w(x, u)$ is a word in x and u, i.e. an element of F_1, and $w(y, t)$ is a word in y and t, i.e. an element of F_2, can hold only if $w(x, u)\in H_1$ and $w(y, t)\in H_2$. It is not hard to see that, in F_1, a word of the form u^ixu^{-j} belongs to H_1 only if $j\in U$. Thus a relation $u^ixu^{-j}=t^iyt^{-j}$ holds in G_U only if $u^ixu^{-j}\in H_1$ holds in F_1, i.e. only if $j\in U$.

CONSTRUCTION. If we now choose U to be a nonrecursive set of integers then the word problem of G_U is not solvable. For if it were solvable then in particular the set of all equations of the form $j^ixu^{-i}=t^iyt^{-i}$ holding in G_U would be recursive, and since such an equation holds if and only if $j\in U$, the set U would be recursive.

1.6. **Computable rings and fields.** An indexing i of a ring R will be called *admissible* if both the corresponding addition function $s(j, k)$ (defined by the condition $a_j+a_k=a_{s(j,k)}$, $j\in i(R)$, $k\in i(R)$) and the corresponding multiplication function $m(j, k)$ are computable. The notions of a computable ring, recursive realization, computable and strongly computable homomorphisms, may now be defined in complete analogy with Definitions 5, 6, and 7. Com-

putable fields are a special case of computable rings. Theorem 1 reads, for the case of rings, as follows: *If i is an admissible indexing of the ring R and H is a two-sided ideal in R such that $i(H)$ is a recursive set of integers, then the quotient ring R/H is computable. Furthermore R/H possesses an admissible indexing i such that the natural homomorphism $\phi: R \to R/H$ is computable with respect to i and i_1.*

To prove this define a function $f: i(R) \to I$ by the condition

$$f(i(a)) = \min_{b \in a+H} i(b).$$

Define a function $i_1: R/H \to I$ by $i_1(a+H) = f(i(a))$. Let $m(j, k)$ and $s(j, k)$ be the multiplication and addition functions corresponding to i. The proof of Theorem 1 now carries over to show that i_1 is an indexing of R/H and that $m_1(j, k) = f(m(j, k))$ and $s_1(j, k) = f(s(j, k))$, where $j \in i_1(R/H)$, $k \in i_1(R/H)$, are the corresponding multiplication and addition functions. Again as in the proof of Theorem 1, f, m_1, and s_1, turn out to be computable functions; thus i_1 is an admissible indexing of R/H. Finally, f is the mapping from $i(R)$ onto $i_1(R/H)$ induced by the natural homomorphism $\phi: R \to R/H$; the homomorphism ϕ is therefore computable with respect to i and i_1.

Theorems 2 and 3 also generalize in a straightforward manner to the case of computable rings.

1.7. **Matrix representations.** An important branch of group theory is the theory of group representations. A *matrix representation* of a group G is a group H of matrices over some field F together with a homomorphism ϕ of G onto H. A group G is said to possess a *faithful matrix representation* (over a field F) if there exists a group H of matrices (over F) such that $G \approx H$. By applying the concept of a computable group we shall prove that certain groups do not possess faithful representations.

In a forthcoming paper we shall prove the following

THEOREM 5. *Every finitely generated group of matrices over any field is a computable group.*

Notice that no assumption concerning the computability of the field is made in the statement of Theorem 5.

THEOREM 6. *A finitely generated group G with a word problem which is not recursively solvable (e.g. the group G_U constructed before) does not possess any faithful representation by matrices over any field.*

Proof. Assume to the contrary that H is a group of matrices over some field such that $H \approx G$. Since G is finitely generated so is H; by the previous theorem H is therefore a computable group. Thus G is isomorphic to a computable group and hence is computable; but the word problem of G is not solvable so that by Theorem 4 G is not computable, a contradiction.

REMARK. Every finitely presented group with a word problem which is

not recursively solvable is also, because of Theorems 4 and 6, an example of a group which does not possess faithful matrix representations. The existence of finitely presented groups with a word problem which is not recursively solvable was proved by Novikov [8] and again by Boone [1].

Examples of finitely generated groups which do not possess faithful matrix representations were constructed, using purely algebraic methods, by Fuchs-Rabinovitch [5; 6].

Chapter II. Computable fields, algebraic closures, and splitting algorithms

2.1. **Terminology and preliminary lemmas.** We now shift our attention to the special case of computable fields and proceed to use the general theory of computable algebraic systems for obtaining specific information about fields.

The definition of the notions of admissible indexing, computable ring (so in particular field), computable and strongly computable homomorphisms, are entirely analogous to Definitions 4–7 and were discussed in §1.6. We recall that for an indexing $i: F \to I$ of a field F to be admissible it is now required that *both* the addition function and the multiplication function corresponding to i (Definition 4) are computable functions.

Before tackling the main theorem of this chapter, let us introduce two lemmas concerning the effectiveness of certain operations in computable fields.

Let F be a computable field and i an admissible indexing of F. Let R be the ring $F[x_1, x_2, \cdots]$ of all polynomials in the countable sequence of indeterminates $x_1, x_2, \cdots$ with coefficients in F. We shall say that an indexing i_0 of a set $S \subseteq R$ of polynomials is *effective* (with respect to i) if $i_0(f)$, where $f \in S$, is effectively computable from the i-indexes of the coefficients of f; and if conversely the number of nonzero coefficients and the i-indexes of the nonzero coefficients of the polynomial $i_0^{-1}(k)$ are effectively computable from the integer $k \in i_0(S)$.

There certainly exist indexings of R itself which are effective with respect to i.

If $S \subseteq R$ is a set of polynomials we shall say that S is *recursive* (with respect to i) if for some (and hence for every) effective indexing i_R of R, the set $i_R(S)$ is a recursive set of integers. The notion of recursiveness of a set extends in an obvious way to sets of finite sequences of polynomials of R.

Let $S \subseteq R$ be a recursive set of polynomials (with respect to i). We shall say that an integral-valued function $\phi: S \to I$ is *computable* (with respect to i) if for some (and hence for every) effective indexing i_0 of S the function $f: i_0(S) \to I$ defined by the relation

$$f(k) = \phi(i_0^{-1}(k)), \qquad k \in i_0(S),$$

is a computable function.

All the notions defined above will also be used when dealing with rings $F[x_1, \cdots, x_n]$ of polynomials in n indeterminates.

LEMMA 4. *Let F be a computable field and i an admissible indexing of F. The function $\nu: F[t] \to I$ assigning to each polynomial $f(t) \in F[t]$ the number of different roots that the equation $f(t) = 0$ has in the algebraic closure of F, is a computable function.*

Proof. Let

$$f(t) = a_0 t^{n_0} + a_1 t^{n_1} + \cdots + a_m t^{n_m}, \quad a_i \neq 0, \quad n_0 > n_1 > \cdots > n_m.$$

Define a polynomial $f_1(t)$ as follows: If the characteristic of F is 0 then let $f_1(t) = f'(t)$ (the formal derivative of $f(t)$). If F is of characteristic $p > 0$ and $n_i = r_i p^e$, where $p^e = q$ is the highest power of p dividing all exponents of $f(t)$, then let

$$f_1(t) = a_0 r_0 t^{n_0 - q} + a_1 r_1 t^{n_1 - q} + \cdots + a_m r_m t^{n_m - q}.$$

It can be shown, but we shall not do it here, that $(f(t), f_1(t)) = 1$ if and only if $f(t)$ has only simple roots, or in the case of characteristic p, only q-fold roots.

We thus have the following effective procedure for computing $\nu(f(t))$. Form $f_1(t)$, i.e. compute the i-indexes of its coefficients. Compute $g(t) = (f(t), f_1(t))$; this requires only rational operations with elements of F and can be done effectively. If $g(t) = 1$ then $\nu(f(t))$ is known. If $g(t) \neq 1$ then $g(t)$ has a smaller degree than $f(t)$ and is a nontrivial factor of $f(t)$, thus $f(t) = q(t)g(t)$. Compute $q(t)$ and $r(t) = (q(t), g(t))$. The degrees of $g(t)$, $q(t)$, and $r(t)$ are less than the degree of $f(t)$. Using an induction hypothesis, $\nu(g(t))$, $\nu(q(t))$, and $\nu(r(t))$ are effectively computable. Now

$$\nu(f(t)) = \nu(g(t)) + \nu(q(t)) - \nu(r(t)).$$

LEMMA 5. *Let F be a computable field and i an admissible indexing of F. Let $R = F[x_1, x_2, \cdots]$. The set U of all finite sequences $(f_1, \cdots, f_n)$, $f_i \in R$, for which there exists a solution $r_1 \in R, \cdots, r_n \in R$, of the equation*

$$r_1 f_1 + \cdots + r_n f_n = 1, \tag{2.1}$$

is a recursive set (with respect to i). In more intuitive language: There exists an effective procedure for determining, when given the i-indexes of the coefficients of polynomials $f_1 \in R, \cdots, f_n \in R$, whether equation (2.1) is solvable.

Proof. Assume that $x_1, \cdots, x_m$ are all the indeterminates appearing in $f_1, \cdots, f_n$, so that $f_i = f_i(x_1, \cdots, x_m)$. It is well known that (2.1) has solutions if and only if the algebraic equations

$$f_1(x_1, \cdots, x_m) = 0, \cdots, f_n(x_1, \cdots, x_m) = 0$$

do not possess a common solution $x_1 = \xi_1, \cdots, x_m = \xi_m$ in the algebraic closure

of F. The question whether such a given system of algebraic equations has a solution can be answered by elimination theory. The instructions for the elimination process are effective and the process involves only rational operations with the coefficients of the given polynomials. We can thus settle the question by performing the elimination computations with the known i-indexes of the coefficients of $f_1, \cdots, f_n$. Since i is an admissible indexing of the computable field F, the whole process is effective.

2.2. **Algebraic closures.** Let F be a field. In the sequel we shall denote the algebraic closure of F by $\overline{F}$. We assume that $\overline{F}$ actually contains F as a subfield and is algebraic over F. The *imbedding isomorphism* of F into $\overline{F}$ is then the mapping ϕ: $F \to \overline{F}$ such that $\phi(b) = b$ for all $b \in F$.

THEOREM 7. *If F is a computable field and i is an admissible indexing of F then the algebraic closure $\overline{F}$ of F is computable and there exists an admissible indexing i_1 of $\overline{F}$ such that he imbedding isomorphism of F into $\overline{F}$ is computable with respect to i and i_1.*

Proof. Form the ring $R = F[x_1, x_2, \cdots]$ by adjoining to F a countable sequence of indeterminates. Let i_R be an effective indexing of R (for the terminology see beginning of §2.1). Since F is computable, i_R is an admissible indexing of R. Furthermore, the imbedding isomorphism of F into R is computable with respect to i and i_R.

The field F possesses an indexing and is therefore at most countable. Let S be the (countable) set of all nonconstant polynomials in $F[t]$. Let i_0 be an effective indexing of S which maps S onto the positive integers. The sequence $\Gamma = f_1(t), f_2(t), \cdots$, (where $i_0(f_j(t)) = j$) thus exhausts S.

Let $n(j)$ be the number of distinct roots which the equation $f_j(t) = 0$ has in the algebraic closure of F. Since the indexing i_0 is effective with respect to i and furthermore i is an admissible indexing of F, the function $n(j)$ is computable (Lemma 4).

With each number $n \geqq 1$ associate the polynomial

$$D_n(y_1, \cdots, y_n, y_{n+1}) = y_{n+1} \cdot \prod_{k \neq j;\, k,j \leqq n} (y_k - y_j) - 1. \tag{2.2}$$

Define now a sequence Γ_1 of polynomials of $F[x_1, x_2, \cdots]$.

$$\Gamma_1 = f_1(x_1), f_1(x_2), \cdots, f_1(x_{n(1)}), D_{n(1)}(x_1, \cdots, x_{n(1)}, x_{n(1)+1}),$$
$$f_2(x_{n(1)+2}), \cdots, f_2(x_{n(1)+n(2)+1}),$$
$$D_{n(2)}(x_{n(1)+2}, \cdots, x_{n(1)+n(2)+1}, x_{n(1)+n(2)+2}), \cdots.$$

Denote the kth element of Γ_1 by $g^{(k)}$. Since i_0 and i_R are effective indexings and $n(j)$ is a computable function (Lemma 4), the function $h(k) = i_R(g^{(k)})$ is computable.

We contend that for every $k \geqq 0$ the equation

$$r^{(1)}g^{(1)} + r^{(2)}g^{(2)} + \cdots + r^{(k)}g^{(k)} = 1$$

does not possess solutions $r^{(1)}, r^{(2)}, \cdots, r^{(k)} \in R$. This is true because the algebraic equations $g^{(1)}=0, g^{(2)}=0, \cdots, g^{(k)}=0$ have a common system of solutions in some extension field of F.

Define now a subset U of R as follows. Assume, in order to simplify notations, that $i_R(R)=I$ (set of all integers); let r_j denote the polynomial of R for which $i_R(r_j)=j$. For every $j \geqq 1$, if $s(1), s(2), \cdots, s(p)$ are all the i_R-indexes, smaller than j, of elements in U, then $r_j \in U$ if and only if for no k the equation

$$r^{(0)}r_j + r^{(1)}r_{s(1)} + \cdots + r^{(p)}r_{s(p)} + r^{(p+1)}g^{(1)} + \cdots + r^{(p+k)}g^{(k)} = 1 \tag{2.3}$$

has a solution $r^{(0)}, r^{(1)}, \cdots, r^{(p+k)} \in R$. This regressive condition determines U completely. The reader can verify that

$$\Gamma_1 \subseteq U, \tag{2.4}$$

$$U \text{ is a nontrivial ideal of } R, \tag{2.5}$$

$$U \text{ is a maximal nontrivial ideal of } R. \tag{2.6}$$

Next we show that $i_R(U)$ is a recursive set of integers. To determine whether $j \in i_R(U)$ we must determine, assuming that we already know all integers $s(1)<s(2)<\cdots<s(p)<j$, that are elements of $i_R(U)$, whether some equation (2.3) has a solution. It is enough to consider a k so large that none of the polynomials $g^{(k+1)}, g^{(k+2)}, \cdots$ contains an indeterminate occurring in one of $r_{s(1)}, r_{s(2)}, \cdots, r_{s(p)}, r_j$; the smallest such k, call it $k(j)$, can be computed effectively. If the equation

$$r^{(0)}r_j + r^{(1)}r_{s(1)} + \cdots + r^{(p)}r_{s(p)} + r^{(p+1)}g^{(1)} + \cdots + r^{(p+k(j))}g^{(k(j))} = 1 \tag{2.7}$$

has a solution then $j \notin i_R(U)$. If this equation does not have a solution then none of the equations (2.3) has a solution and thus $j \in i_R(U)$. The question whether equation (2.7) has a solution $r^{(0)} \in R, r^{(1)} \in R, \cdots, r^{(p+k(j))} \in R$ can now be settled effectively (Lemma 5).

We thus have the following effective procedure for determining whether $j \in i_R(U)$. Determine whether $1 \in i_R(U)$ (i.e. whether $r_1 \in U$) by the above method. After knowing whether $1 \in i_R(U)$ or not, proceed to determine by the above method whether $2 \in i_R(U)$. Proceed in this way until j is reached, by that time all $s<j$ for which $s \in i_R(U)$ are known; the above method can therefore be applied to determine whether $j \in i_R(U)$. This procedure is effective; the set $i_R(U)$ is therefore recursive.

By §1.6 the quotient ring R/U is computable and possesses an admissible indexing i_1 such that the natural homomorphism $\phi: R \to R/U$ is computable with respect to i_R and i_1. Let us recall that $F \subset R$ and that the imbedding isomorphism $\psi: F \to R$ is computable with respect to i and i_R. Because of (2.5) we have $U \cap F = (0)$, ϕ therefore acts on F as an isomorphism and we can identify $\phi(F) \subseteq R/U$ with F. The imbedding isomorphism $\phi\psi: F \to R/U$, being

the product of two computable homomorphisms, is computable with respect to i and i_1.

All that remains is to show that R/U is the algebraic closure of its subfield F. Since U is, by (2.6), a maximal nontrivial ideal of R, the ring R/U is a field. If $f(t)$ is any polynomial with coefficients in F then it appears in Γ, say it is $f_j(t)$. The polynomials

$$f_j(x_{k+1}), f_j(x_{k+2}), \cdots, f_j(x_{k+n(j)}), \quad D_{n(j)}(x_{k+1}, \cdots, x_{k+n(j)}, x_{k+n(j)+1})$$

where $k=n(1)+\cdots+n(j-1)+j-1$, are since $\Gamma_1 \subseteq U$, in U. The elements $\phi(x_{k+1}), \cdots, \phi(x_{k+n(j)})$ of R/U are therefore roots of the equation $f_j(t)=0$. Because of the special form (2.2) of $D_{n(j)}$ we have

$$\phi(x_{k+n(j)+1}) \cdot \prod_{p \neq q;\, p,q \leq n(j)} [\phi(x_{k+p}) - \phi(x_{k+q})] - 1 = 0$$

in R/U. The elements $\phi(x_{k+p})$, $p \leqq n(j)$, are therefore pairwise distinct. Thus every polynomial $f_j(t) \in F[t]$ splits into linear factors in the field R/U. Finally let us note that $R/U = F(\phi(x_1), \phi(x_2), \cdots)$ and that the elements $\phi(x_1)$, $\phi(x_2), \cdots$, are algebraic over F. This implies that R/U is indeed the algegraic closure $\overline{F}$ of F. Q.E.D.

2.3. **Splitting algorithms.** A computable field F has a splitting algorithm (with respect to an admissible indexing i) if, roughly speaking, there exists an effective procedure for deciding, for every given polynomial $f(t) \in F[t]$, whether $f(t)$ splits into a nontrivial product $f(t)=p(t)q(t)$, where $p(t)$, $q(t) \in F[t]$. It is readily seen that if F has a splitting algorithm then it is also possible to compute effectively the product decomposition into prime factors of every polynomial $f(t) \in F[t]$. We shall use our previous results to obtain a necessary and sufficient condition for the existence of a splitting algorithm. From this condition we shall then be able to infer the existence of splitting algorithms in some important cases.

DEFINITION 9. A field F is said *to have a splitting algorithm with respect to an admissible indexing i* if the set S of all polynomials $f(t) \in F[t]$ which split into a nontrivial product, is recursive with respect to i (see beginning of §2.1.).

The following lemma will be needed in the proofs of Theorem 8 and, later on, Theorem 9. Its content is, in essence, that if F has a splitting algorithm with respect to an indexing i, then the degree of every element algebraic over F is effectively computable.

LEMMA 6. *Let F be a computable field and i an admissible indexing of F. Let i_1 be an admissible indexing of the algebraic closure $\overline{F}$ and let the imbedding isomorphism ϕ: $F \to \overline{F}$ be computable with respect to i and i_1. If F possesses a splitting algorithm with respect to i, then the function d: $i_1(\overline{F}) \to I$ defined by the condition*

$$d(i_1(\alpha)) = [F(\alpha): F] \quad \textit{for all } \alpha \in \overline{F}, \tag{2.8}$$

is effectively computable.

Proof. To compute $d(j)$ for any given $j \in i_1(\overline{F})$ proceed as follows. Enumerate one by one all sequences $(a_1, \cdots, a_n)$, $a_k \in i(F)$, n runs through the integers, for which the polynomial

$$t^n + i^{-1}(a_1)t^{n-1} + \cdots + i^{-1}(a_n) \in F[t] \tag{2.9}$$

is irreducible. Since F possesses a splitting algorithm the set of these sequences is recursive and hence certainly recursively enumerable (Definition 2) so that the enumeration can be carried out in an effective way. For each such sequence compute the sequence $(\bar{\phi}(a_1), \cdots, \bar{\phi}(a_n))$ where $\bar{\phi}: (F, i) \to (\overline{F}, i_1)$ is the isomorphism induced by the imbedding $\phi: F \to \overline{F}$. Since $\bar{\phi}$ is a computable function this is effective. Next compute

$$j^{(n)} + \bar{\phi}(a_1)j^{(n-1)} + \cdots + \bar{\phi}(a_n) \tag{2.10}$$

where addition, multiplication, and exponentiation (k), are the operations of $(\overline{F}, i_1)$; this again can be done effectively. After a finite number of such steps a sequence $(a_1, \cdots, a_n)$ will be reached for which the expression (2.10) will reduce to the i_1-index of the zero element of $\overline{F}$. Equation (2.9) is then the irreducible equation of $i_1^{-1}(j)$ over F. The number n of elements in the sequence therefore satisfies $n = d(j)$. The above given instructions and computations for finding this n are effective so that d is indeed a computable function.

THEOREM 8. *Let F be a computable field and i an admissible indexing of F. The field F has a splitting algorithm with respect to i if and only if the algebraic closure $\overline{F}$ has an admissible indexing i_1 such that the imbedding isomorphism of F into $\overline{F}$ is strongly computable with respect to i and i_1 (Definition 7).*

Proof. By Theorem 7 there exists an admissible indexing, call it i_1, of $\overline{F}$ such that the imbedding isomorphism $\phi: F \to \overline{F}$ is computable with respect to i and i_1. Assume first that F possesses a splitting algorithm with respect to i and let us show that ϕ is strongly computable with respect to i and i_1. By Lemma 6 the function d defined by (2.8) is computable. Now if $j \in i_1(\overline{F})$, then $j \in i_1(F)$ if and only if $d(j) = 1$. We thus have an effective test for membership in $i_1(F)$ so that this set is recursive. This means that ϕ is strongly computable with respect to i and i_1.

Assume next that $\overline{F}$ has an admissible indexing i_1 such that the imbedding isomorphism $\phi: F \to \overline{F}$ is strongly computable; we proceed to prove that F has a splitting algorithm with respect to i. Let $\bar{\phi}: (F, i) \to (\overline{F}, i_1)$ be the isomorphism induced by ϕ, then $\bar{\phi}$ is a computable function and the set $\bar{\phi}(i(F)) = i_1(\phi(F))$ is, by assumption, a recursive set of integers.

Given any sequence $(a_1, \cdots, a_n)$, $a_k \in i(F)$ apply the following effective procedure to determine whether the polynomial

$$f(t) = t^n + i^{-1}(a_1)t^{n-1} + \cdots + i^{-1}(a_n)$$

splits in $F[t]$. Compute the integers $\bar{\phi}(a_1), \cdots, \bar{\phi}(a_n)$. By substituting successively all integers $j \in i_1(\overline{F})$, computing (in $(\overline{F}, i_1)$) the expression,

$$j^{(n)} + \bar{\phi}(a_1) \cdot j^{(n-1)} + \cdots + \bar{\phi}(a_n)$$

and seeing whether it reduces to the i_1-index of the zero element of $\overline{F}$, find an i_1-index of a root α_1 of the equation $f(t) = 0$. Next compute (in $(\overline{F}, i_1)$) the i_1-indexes of the coefficients of the polynomial $f_1(t) = f(t)(t - \alpha_1)^{-1}$. Repeat the previous procedure to find a root of the equation $f_1(t) = 0$. Continue this process until all i_1-indexes, $k_1, k_2, \cdots, k_n$, of roots of the equation $f(t) = 0$ are found. Then compute in $(\overline{F}, i_1)$ all expressions

$$(t - k_{j_1}) \cdots (t - k_{j_r})$$

where $(j_1, \cdots, j_r)$ runs through all *proper* subsets of $\{1, \cdots, n\}$. If, after multiplying out, all coefficients of one of these expressions are in the set $i_1(\phi(F)) = \bar{\phi}(i(F))$ then $f(t)$ has a proper factor in $F[t]$. Whether the aforementioned coefficients are in the set $i_1(\phi(F))$ can be effectively checked because this set is recursive. Thus the whole process is effective and F has a splitting algorithm with respect to i.

2.4. **A theorem of van der Waerden.** Van der Waerden proved [12, pp. 134–137] that if F is a computable field with a splitting algorithm and α is algebraic and separable over F then the field $F(\alpha)$ has a splitting algorithm. Fröhlich and Shepherdson showed by means of an example [4] that the condition that α is separable cannot be omitted. Our general theorems yield a very simple and natural proof for the van der Waerden theorem. Let us first of all give a precise formulation.

THEOREM 9 (VAN DER WAERDEN). *Let i be an admissible indexing of the computable field F, and let F possess a splitting algorithm with respect to i. If α is algebraic and separable over F then the extension field $F(\alpha)$ possesses an admissible indexing i_0 such that* (a) *the imbedding isomorphism $\psi: F \to F(\alpha)$ is strongly computable with respect to i and i_0,* (b) *the field $F(\alpha)$ possesses a splitting algorithm with respect to i_0.*

Proof. Let $\overline{F}$ denote the algebraic closure of F. By Theorem 8 there exists an admissible indexing, call it i_1, of $\overline{F}$ such that the imbedding isomorphism $\phi: F \to \overline{F}$ is strongly computable with respect to i and i_1. The induced isomorphism $\bar{\phi}: (F, i) \to (\overline{F}, i_1)$ is thus a computable function.

According to our conventions F is a subset (subfield) of $\overline{F}$; thus also $F(\alpha) \subseteq \overline{F}$. We contend that $i_1(F(\alpha))$ is a recursive set.

Let us recall (see [12, pp. 126–127]) that if α is separable over F and $\beta \in \overline{F}$ is any algebraic element then the field $F(\alpha, \beta)$ can be generated by a

single algebraic element, i.e. there exists a $\gamma \in \overline{F}$ such that $F(\alpha, \beta) = F(\gamma)$. In fact, if $\alpha_1 = \alpha, \alpha_2, \cdots, \alpha_n$, and $\beta_1 = \beta, \beta_2, \cdots, \beta_m$, are all the elements of $\overline{F}$ conjugate (over F) to α and β respectively, and if $\gamma = \beta + c\alpha$ where $c \in F$ and $c \neq (\beta - \beta_j)(\alpha_k - \alpha)^{-1}$ for all $j \leqq m$, $k \leqq n$, then $F(\alpha, \beta) = F(\gamma)$. Furthermore we have that $\beta \in F(\alpha)$ if and only if $[F(\alpha, \beta): F] = [F(\alpha): F]$.

We now have the following effective procedure for deciding for each j whether $j \in i_1(F(\alpha))$. Let $i_1(\alpha) = k$. Compute $d(k)$ and $d(j)$; all computation with d are effective since this function is computable (Lemma 6). We may assume that F is infinite since the theorem is trivially true when F is finite. Denote $d(k)d(j)$ by p and let $c_1, c_2, \cdots, c_p$ be the first, in order of magnitude, p integers in $i_1(F)$. Since $i_1(F)$ is a recursive set these integers can be effectively computed. Compute $b_r = j + c_r k$ (where addition and multiplication are the operations of $(\overline{F}, i_1)$) and then compute $d(b_r)$. By the previous paragraph at least one of the elements $i_1^{-1}(b_r)$ generates the field $F(\alpha, i^{-1}(j))$. Thus $j \in i_1(F(\alpha))$ if and only if $d(b_r) \leqq d(k)$ (i.e. $d(b_r) \leqq [F(\alpha): F]$) for $1 \leqq r \leqq p$. We thus have an effective test for membership in $i_1(F(\alpha))$ and this set is recursive.

The restriction of the mapping $i_1: \overline{F} \to I$ to the set $F(\alpha) \subseteq \overline{F}$, call it i_0, is therefore an indexing of $F(\alpha)$. Since i_1 is an admissible indexing of $\overline{F}$ the restriction i_0 is an admissible indexing of $F(\alpha)$. The imbedding isomorphism $\phi: F \to \overline{F}$ is strongly computable with respect to i and i_1. The imbedding isomorphism $\psi: F \to F(\alpha)$ is actually the same function as ϕ so that ψ is strongly computable with respect to i and the restriction i_0; thus (a) is established. The isomorphism $(F(\alpha), i_0) \to (\overline{F}, i_1)$ induced by the imbedding $F(\alpha) \to \overline{F}$ is the identity mapping. The imbedding isomorphism $F(\alpha) \to \overline{F}$ is therefore strongly computable with respect to i_0 and i_1. By Theorem 8 the field $F(\alpha)$ possesses a splitting algorithm with respect to i_0 which proves (b).

2.5. **Concluding remark.** The theory presented in this paper can be generalized as follows. Let C be any class of functions from integers to integers such that if $f_1, f_2, \cdots, f_n \in C$ and if f is recursive in $f_1, f_2, \cdots, f_n$, then $f \in C$. We can now replace everywhere in this paper the phrase "f is computable" by "f is in C" (that is, consider C as the class of all computable functions). All of our definitions and theorems will carry over to this C-theory. Thus a C-indexing of a set S is a one to one function $i: S \to I$ such that the characteristic function of $i(S) \subseteq I$ belongs to C; a C-indexing $i: G \to I$ of a group G is C-admissible if the corresponding multiplication function m satisfies $m \in C$, and so forth.

The question whether the notion of computability is really identical with that of recursiveness is of course nonmathematical and cannot be settled by a mathematical proof. It is simply a matter of definition that mathematicians agree to call a function computable if and only if it is recursive. By the previously remark our results will carry over no matter what notion of computability we adopt as long as the closure condition on the class C of all "computable" functions holds.

BIBLIOGRAPHY

1. W. W. Boone, *Certain simple unsolvable problems of group theory*. V–VI, Nederl. Akad. Wetensch. Proc. ser. A vol. 60 (1957) pp. 22–27; 227–232.

2. N. Bourbaki, *Elements de Mathématique*, Part I, Book 2, Chapters 4–5, Paris, Hermann, 1950.

3. A. Fröhlich and J. C. Shepherdson, *On the factorization of polynomials in a finite number of steps*, Math. Z. vol. 62 (1955) pp. 331–334.

4. ———, *Effective procedures in field theory*, Philos. Trans. Roy. Soc. London ser. A vol. 284 (1955) pp. 407–432.

5. D. I. Fuchs-Rabinowitsch, *Über eine Gruppe mit endlichvielen Erzeugenden und Relationen die keine isomorphe Darstellung durch Matrizen von endlicher Ordnung zulässt*, Dokl. Akad. Nauk SSSR vol. 27 (1940) pp. 425–426.

6. ———, *Beispiel einer diskreten Gruppe mit endlichvielen Erzeugenden und Relationen, die kein vollständiges System der linearen Darstellungen zulässt*, Dokl. Akad. Nauk SSSR. vol. 29 (1940) pp. 549–550.

7. S. C. Kleene, *Introduction to metamathematics*, New York, Van Nostrand, 1952.

8. P. S. Novikov, *On the algorithmic unsolvability of the word problem in group theory* (Russian), Trudy Mat. Inst. Steklov. vol. 44 Izdat. Akad. Nauk SSSR, Moscow, 1955.

9. M. O. Rabin, *Recursive unsolvability of group theoretic problems*, Ann. of Math. vol. 67 (1958) pp. 172–194.

10. H. G. Rice, *Recursive and recursively enumerable orders*, Trans. Amer. Math. Soc. vol. 83 (1956) pp. 277–300.

11. B. L. van der Waerden, *Eine Bemerkung über die unzerlegbarkeit von Polynomen*, Math. Ann. vol. 102 (1930) pp. 738–739.

12. ———, *Modern algebra*, vol. I, New York, Ungar, 1949.

PRINCETON UNIVERSITY,
PRINCETON, NEW JERSEY