BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 6, Number 1, January 1982

THE WORD PROBLEM AND THE ISOMORPHISM PROBLEM FOR GROUPS

BY JOHN STILLWELL

If the fundamental problem of mathematics is to decide when two things are the same, then the fundamental problem of group theory is to decide when two groups are isomorphic. This problem was first stated, for finitely presented groups, by Tietze [**1908**], and proved unsolvable by Adian and Rabin 50 years later. Using their result, Markov [**1958**] proved the unsolvability of the fundamental problem of topology; the homeomorphism problem. Of course, combinatorial group theory and topology grew up together, and their connection via the fundamental group was well known; the bridge between them and logic is the word problem for groups, proved unsolvable by Novikov in 1955.

The history of the word problem divides naturally into three eras: 1880–1930, in which combinatorial group theory interacts mainly with topology and the major positive results are obtained; 1930–1955, in which computability theory emerges and, after a great struggle, yields Novikov's unsolvability proof; 1955–1980, in which group theory interacts with logic to simplify the proof. In 1955, the characteristic properties of groups appeared mainly as obstacles to an unsolvability proof-witness the 143 pages of Novikov's paper. It took 25 years to properly understand the group theoretic construction, the HNN extension, which allows group theory and computation to work together. Today it is clear that the negative theorem on the word problem has brought positive benefits to group theory in the form of techniques suitable for giving a clear proof.

The main purpose of this paper is to give such a proof, based on that of Cohen and Aanderaa [**1980**], but with the historical background necessary for full motivation and understanding. I shall therefore discuss the story of the word problem up to Magnus' solution for one-relator groups around 1930, the notion of computation developed by logicians of the 1930's, results on semigroups which foreshadowed those on groups, before treating the development of HNN theory and its contribution to the word and isomorphism problems.

The technical details have been concentrated in §§1, 4, 6–8, 11–14, so readers who want an unadulterated proof of unsolvability of the word problem need read only these.

1. Review of combinatorial group theory. For logicians, the most natural approach to combinatorial group theory is that of Magnus, used in Magnus

Received by the editors November 15, 1980.

1980 *Mathematics Subject Classification.* Primary 01A60, 20F05, 20F10; Secondary 03D10.

[1930] to define equivalence of words, and developed fully in Magnus, Karrass and Solitar **[1966]**. (Magnus credits the idea to Dehn.)

Generators are letters $a_1, a_2, \ldots,$ and each generator a_i has a *formal inverse* a_i^{-1}, also regarded as a generator.

A *word* is a sequence $a_{i_1}^{\varepsilon_1} \cdots a_{i_k}^{\varepsilon_k}$ of generators, where each $\varepsilon_i = \pm 1$. The empty word is denoted by 1. If u and v are words then uv denotes the *concatenation* of u and v—the sequence which results when u is followed by v.

A pair $\langle a_1, a_2, \ldots ; r_1, r_2, \ldots \rangle$, the first member of which is a set of generators, and the second member of which is a set of words (called *relators* in this context), is called a *presentation*.

We determine a group G from a presentation, and by abuse of language write

$$G = \langle a_1, a_2, \ldots ; r_1, r_2, \ldots \rangle,$$

as follows.

(a) Words w_1, w_2 are called *equivalent* if w_1 can be converted into w_2 by a finite sequence of the following types of transformation:

$$\text{replace } uv \text{ by } ua_i a_i^{-1} v, \qquad \text{replace } uv \text{ by } ua_i^{-1} a_i v,$$
$$\text{replace } uv \text{ by } ur_j v,$$

or their inverses. We also say that the equation $w_1 = w_2$ is a *consequence* of the relations $r_j = 1$ in this case.

(b) The elements of G are the equivalence classes of words. The equivalence class of w is denoted by $[w]$, but we often allow w to stand for $[w]$, (just as in ordinary mathematics we speak of the "rational number $\frac{1}{2}$" when we really mean the equivalence class $\{\frac{1}{2}, \frac{2}{4}, \frac{3}{6}, \ldots\}$).

(c) The *product* of $[u]$, $[v]$ is $[uv]$.

(d) The *identity* of G is $[1]$, which we also write 1.

(e) The *inverse* of $[a_{i_1}^{\varepsilon_1} \cdots a_{i_k}^{\varepsilon_k}]$ is $[a_{i_k}^{-\varepsilon_k} \cdots a_{i_1}^{-\varepsilon_1}]$.

It is now easy to prove that G is indeed a group. The only step not completely routine is to show that the product is well defined. Here one has to observe that transformations which change, say, u into another representative u' will likewise change uv into $u'v$, hence the product is independent of the choice of representatives.

If the sets $\{a_i\}$ and $\{r_j\}$ are finite in some presentation of G then G is said to be *finitely presented*. Given a fixed finite presentation of G, its *word problem* is the problem of deciding, for any word w, whether $w = 1$ in G. The problem of deciding whether two words u, v are equal of course reduces to this, namely, ask whether $uv^{-1} = 1$. We state the word problem only for finitely presented groups, since this is the interesting case, and anyway it is not clear what it means to be "given" an infinitely presented group.

The *isomorphism problem* for finitely presented groups is the problem of deciding, for any two finite presentations, whether they represent isomorphic groups.

2. Origins of the word problem. Combinatorial group theory emerged from complex function theory and topology in the 1880's, with the work of Klein, Fricke and Poincaré (see Stillwell [**1980**] or Magnus and Chandler (to appear)). Poincaré [**1892**, **1895**] introduced the fundamental group to solve problems in topology, but quickly realized that it was a two-edged sword; as much as group theory made topology easier, so too did topology make group theory harder, by extending its range of application to problems of unprecedented difficulty.

The problem which led to the first statement of the word problem, by Dehn [**1910**], is that of deciding when two knots are the same. Leaving aside a few subtleties such as right- and left-handedness, this is essentially a homeomorphism problem, namely: given two knots K_1 and K_2 in $\mathbf{R}^3$, decide whether the knot complements $\mathbf{R}^3 - K_1$ and $\mathbf{R}^3 - K_2$ are homeomorphic. Assuming that the knots K_1 and K_2 are polygonal, this is by any reasonable standards a question about *finite* objects (e.g., replacing $\mathbf{R}^3$ by a cube and drilling out a tubular neighbourhood of the knot, one can subdivide the knot complement into finitely many tetrahedra). Thus for the first time, it would seem, mathematicians found themselves unable to decide in general when two finite objects were the same.

Dehn was led to the word problem by considering the special case of deciding whether a given knot K is trivial. He discovered the remarkable result that this is so if and only if the fundamental group $\pi_1(\mathbf{R}^3 - K)$ is abelian (in which case it is obviously infinite cyclic), and this in turn can be decided from a solution to the word problem for $\pi_1(\mathbf{R}^3 - K)$. Dehn's argument for the latter step involves a little more topology, but in fact it is an obvious consequence: a group with generators $a_1, \ldots, a_n$ is abelian if and only if each a_i, a_j commute, and this can be decided by using the solution of the word problem to check whether the finitely many words $a_i a_j a_i^{-1} a_j^{-1}$ all equal 1.

It was already known, implicitly in Poincaré [**1895**] and explicitly in Tietze [**1908**], that a topological space which is finitely presented in terms of cells (vertices, edges, faces, etc.) has a finitely presented fundamental group, so one is led to expect difficulties with finitely presented groups, because of the extent to which they reflect difficult topological problems. Tietze expressed pessimism about the isomorphism problem in his 1908 paper. Dehn extended this pessimism to the seemingly more elementary level of the word problem, but of course neither of them knew then that unsolvability could be formalized or proved to exist.

3. Positive solutions of the word problem. When Dehn stated the word problem in 1910 he was able to point to a special case where the solution was well known: the fundamental groups $\pi_1(S)$ of closed orientable surfaces S. The geometric interpretation of $\pi_1(S)$, in which generators are closed cuts which reduce S to a polygon and the defining relator is the sequence of edges in the polygon boundary, reduces the word problem to a topological problem which had been solved 30 years earlier. This problem is to decide whether a given closed path w (represented by a word on the generators) contracts to a point on S, and it is solved by constructing the universal covering surface $\tilde{S}$, by pasting copies of the polygon for S together (due to Schwarz in 1882, see

Klein [**1882**]). Then w "lifts" to a path $\tilde{w}$ on $\tilde{S}$ described by the same edge sequence, and w is contractible on S if and only if $\tilde{w}$ is closed.

Dehn's contribution was to realise that the pattern of edges in $\tilde{S}$ formed a diagram of $\pi_1(S)$ whose properties implied a *purely algebraic* algorithm for the word problem. Thus one is spared the actual construction of $\tilde{S}$, which is very difficult for an S of genus > 1, and one can operate directly on the word w. Dehn's algorithm is to repeatedly shorten w by cancellation of terms $a_i a_i^{-1}$ or $a_i^{-1} a_i$ and replacement of subwords which are more than half the defining relator by their shorter complements (viewing the relator as a circular word); $w = 1$ if and only if w can be reduced to 1 by this process.

In general, the word problem for the fundamental group $\pi_1(C)$ of any finite complex C could be solved by constructing the universal covering $\tilde{C}$ of C and tracing the path $\tilde{w}$ in $\tilde{C}$ corresponding to the word w. However, $\tilde{C}$ is very hard to find even when C is a knot complement. Dehn solved the word problem only for the trefoil knot group, though he was able to make a wonderful application of this result, proving that the right and left trefoil knots are distinct in Dehn [**1914**]. Another beautiful example of a geometrically motivated word problem with an unexpected algebraic solution is that for braid groups, in Artin [**1926**].

However, these examples fill a very small space in the panorama of finitely presented groups. The first solution of the word problem for a really broad class of groups was that for one-relator groups by Magnus [**1932**]. The proof is difficult, and it has not been essentially simplified or varied since Magnus found it. Furthermore, no other solutions of comparable generality have been found since 1932, though there has been progress with groups of topological interest. In particular, Waldhausen [**1968**] solved the word problem for all knot groups.

4. Free groups. The group with n generators and no relators,

$$F_n = \langle a_1, \ldots, a_n; - \rangle$$

is called the *free group* of *rank n*. We say that $a_1, \ldots, a_n$ freely generate F_n, or form a *basis* for F_n, because they are subject to no relations other than those true in any group, namely $a_i a_i^{-1} = a_i^{-1} a_i = 1$. An example of a generating set for F_n which is not a basis is $\{a_1, \ldots, a_n, a_1 a_2\}$, in which the last generator is the product of the first two. Rank is meaningful because any basis for F_n has n elements, as can be seen by taking the abelian quotient of F_n which results from the relations $a_i a_j = a_j a_i$ and appealing to the well-known invariance of rank for abelian groups.

The word problem for F_n is solved by the process of *free reduction*. Given a word w, one repeatedly cancels subwords of the form $a_i a_i^{-1}$ or $a_i^{-1} a_i$ until none remain. The resulting word is unique and called *freely reduced*. Then $w = 1$ if and only if its freely reduced form is empty.

Nielsen [**1921**] gave a process for reducing any set $\{u_1, \ldots, u_m\}$ of words in F_n to a basis for the subgroup they generate, showing in particular that the subgroup is free and also giving an algorithm which decides whether a given word w belongs to it. We shall not need the Nielsen process, but we do use one example where this type of result can be proved using free reduction.

PROPOSITION. (i) *Let $F = \langle x, y, z; - \rangle$ and let $u_1 = x^m$, $u_2 = x^i z y^j$, $u_3 = y^n$, where $m, n \neq 0$ and i, j are any integers. Then $\{u_1, u_2, u_3\}$ is a basis for the subgroup of F it generates.*

(ii) *If $x^{i_1} z y^{j_1} \in \langle u_1, u_2, u_3; - \rangle$ then $i_1 = i + m_1 m$, $j_1 = j + n_1 n$ for some integers m_1, n_1.*

PROOF. (i) Suppose we have a word w on $\{u_1, u_2, u_3\}$, freely reduced with respect to u_1, u_2, u_3 by cancellation of subwords $u_i u_i^{-1}$ or $u_i^{-1} u_i$. If no terms u_2 occur then w is clearly a freely reduced word on x, y, hence if $w = 1$ it is empty by the solution of the word problem for F.

If $u_2 = x^i z y^j$ occurs, then I claim that no two consecutive occurrences of it can cancel. This is certainly true if their exponents have the same sign, and if they have opposite signs the space between the z's looks like

(a) $z y^n W y^{-n} z^{-1}$ or (b) $z^{-1} x^{-m} W x^m z$ where W is a freely reduced nonempty word on $u_1 = x^m$, $u_3 = y^n$. In case (a) the word $y^n W y^{-n}$ between the z's can only disappear if its exponent sum for y is 0, hence if the exponent sum for y in W is 0. Then since W is freely reduced it must contain x's, and since these do not cancel in W they do not cancel in $y^n W y^{-n}$ either. The situation is similar in case (b), thus the space between any two z's cannot be emptied, and hence $w \neq 1$ if its reduced form on u_1, u_2, u_3 is nonempty.

This says that no nontrivial relation holds between u_1, u_2, u_3, and hence they are a basis for the subgroup they generate.

(ii) If $x^{i_1} z y^{j_1}$ equals a freely reduced word w on u_1, u_2, u_3 then w must have exactly one occurrence of $u_2 = x^i z y^j$, since one is needed to produce the z in $x^{i_1} z y^{j_1}$, and more than one will not cancel, by the above argument. Similarly, no $u_3 = y^n$ can occur to the left of u_2 in w, nor $u_1 = x^m$ to the right, since there is no y to the left of z in $x^i z y^j$, nor x to the right.

Thus $w = u_1^{m_1} u_2 u_3^{n_1}$ for some integers m_1, n_1 i.e. $x^{i_1} z y^{j_1} = x^{i + m_1 m} z y^{j + n_1 n}$ and since both sides are freely reduced words on x, y, z we have $i_1 = i + m_1 m$, $j_1 = j + n_1 n$. □

5. Computability theory in the 1930's. In his celebrated paper on the incompleteness of Principia Mathematica and related systems, Gödel **[1931]** showed that the symbol manipulations of formal logic could be simulated by functions on the natural numbers which he called "rekursiv". These are functions built from very simple, and obviously computable, functions such as constants and successor by equations which define new values in terms of values previously obtained. For example, from the successor function s one can define + by the equations

$$x + 0 = x, \qquad x + s(y) = s(x + y),$$

then × from the equations

$$x \times 0 = 0, \qquad x \times s(y) = (x \times y) + x,$$

and so on. The possibility that any computable function might be obtainable by recursions did not occur to Gödel, though his paper provides evidence for this hypothesis in showing that operations not apparently related to recursion, namely the symbol manipulations of logic, are in fact so definable.

Independently of Gödel, Church was lecturing at Princeton in 1931 on his own system for defining functions, called the λ-calculus. The system was intended to clarify the meaning of variables, not computation, but when news of the "rekursiv" functions arrived, Church set his Ph.D. student Kleene the task of finding which functions were λ-definable. Kleene relates, in Crossley **[1974]** and Kleene **[1979]**, that he got stuck almost immediately on the predecessor function: $f(0) = 0$, $f(s(x)) = x$. The solution came to him in a dentist's office while waiting to have two wisdom teeth pulled, and after this breakthrough he needed only 5 or 6 months to λ-define all the computable functions he and Church could think of.

By 1933, Church was convinced that λ-definability was an exact equivalent of the intuitive notion of computability, and proposed making it the definition. This proposal is now known as *Church's thesis*. Gödel came to the Institute of Advanced Study in 1933, but did not accept Church's thesis at first, proposing instead a more precise formulation of his notion of "rekursiv", called "general recursive".

In 1936 the notions of λ-definability and general recursiveness were proved to coincide, but by that time Gödel had been convinced of the correctness of Church's thesis by the *Turing machine* concept of computability, introduced by Turing **[1936]**.

Turing arrives at his concept by stripping inessentials from the process of computation as experienced by a human being. All that really matter are the eye which scans and recognises symbols, the hand which writes, and the mental states which direct the actions of eye and hand. Assuming that sufficiently similar symbols or mental states will not be distinguished, Turing concludes that computation requires only finitely many mental states and a finite alphabet of symbols. Further, we can place the symbols in individual cells and scan only one cell at a time, because de facto scanning of larger sets of cells is always obtainable by "remembering" (by mental state) a finite number of previously scanned symbols and their positions. Similarly, scanning movements can be broken down into steps from cell to adjacent cell. Finally, we can take the cells to be squares on an infinite strip of tape, since any higher-dimensional array of cells can only be traversed by coordinatising the cells in some computable way, and these coordinates can equally well be assigned to cells on the infinite tape.

In short, a computation is determined by a finite set of symbols $\{S_0, S_1, \ldots, S_m\}$, a finite set of *internal* ("mental") *states* $\{q_0, q_1, \ldots, q_n\}$ and a *response function* $f(q_i, S_j)$ which gives the action to be performed when the internal state is q_i and the scanned symbol is S_j. The only possible responses are of the forms:

Replace S_j by $S_{j'}$, and go into state $q_{i'}$.
Move one cell to right and go into state $q_{i'}$.
Move one call to left and go into state $q_{i'}$.
Halt.

The distillation of the Turing machine concept from intuitive notions made it extremely plausible that the idea of computation had been completely captured, but other evidence was also available. First, equivalence with general recursiveness and λ-definability could be proved. Second, an independent discovery of the Turing machine concept was made by Post [**1936**].

Since 1936, Church's thesis has been accepted and used in the following way to prove unsolvability results. A problem P is given which consists of infinitely many questions Q_i. Solution of P is viewed as the computation of a function

$$f(Q_i) = \begin{cases} \text{YES} & \text{if the answer to } Q_i \text{ is YES,} \\ \text{NO} & \text{if the answer to } Q_i \text{ is NO.} \end{cases}$$

Next one shows that no such function f is computable by Turing machine, i.e. that P is unsolvable by Turing machine. Church's thesis then implies that P is absolutely unsolvable.

The first such applications of Church's thesis were in areas closest to its origin: formal logic, formal number theory and the theory of Turing machines. But as early as 1936 Church speculated (see Kleene [**1979**]) that problems not obviously related to computation might be proved unsolvable, and in 1938 (review in the Journal of Symbolic Logic, p. 74) he explicitly stated the knot problem and the word problem for groups as candidates.

6. Turing machines. Following Turing [**1936**], we reduce the various possible responses of a machine to a given situation (q_i, S_j) = (internal state, scanned symbol) to just two types:

(i) change S_j to $S_{j'}$, move one cell to right, and go into state $q_{i'}$; or

(ii) change S_j to $S_{j'}$, move one cell to left, and go into state $q_{i'}$.

Of course, moving without changing the symbol is included by the possibility $S_j = S_{j'}$, and changing the symbol without moving is the end result of suitable moves to right and left in succession. If no response is specified for a pair (q_i, S_j), then the machine is understood to halt when it reaches this situation.

The response function $f(q_i, S_j)$, which completely determines a machine M, can then be viewed as a finite list of quintuples $q_iS_jS_{j'}Dq_{i'}$, where q_iS_j is the situation, $S_{j'}$ the symbol which replaces S_j, $D = R$ or L is the direction of movement, and $q_{i'}$ the new internal state. The behaviour of M is completely determined by its *input* (the expression initially on the tape, with the scanned symbol distinguished) and the initial state, provided we assume that there is a unique response to each situation (q_i, S_j), i.e. that q_iS_j begins at most one quintuple.

To give a simple example, the following machine

$$q_0\,1\,1\,R\,q_0, \qquad q_0\,\square\,1\,R\,q_1,$$

when placed anywhere on a block of 1's, will travel to the right hand end, insert a 1 in the first blank cell, □, then halt.

This formulation of Turing machines assumes that the tape is infinite in both directions, so that the machine can always move left or right with no danger of falling off. We now give a numerical formulation of Turing machines, evolved from Minsky [**1961**] and Cohen and Aanderaa [**1980**], which provides additional tape out of sheer nothing.

At any instant the future of M's computation is determined by the tape expression, scanned cell and internal state, which we can identify with the following word, called the *complete state*:

$$S_{k_u} \cdots S_{k_2}S_{k_1}q_iS_{j_1}S_{j_2} \cdots S_{j_v}$$

where $S_{k_u} \cdots S_{k_2}S_{k_1}S_{j_1}S_{j_2} \cdots S_{j_v}$ = tape expression, q_i = current internal state, S_{j_1} = scanned symbol. The computation itself can then be identified with the sequences of complete states produced by successive responses of M, the transformation of one complete state to its successor being called a *step* of computation.

Now we construct the pair, called the *complete state pair*,

$$(S_{k_u} \cdots S_{k_2}S_{k_1}q_i, S_{j_v} \cdots S_{j_2}S_{j_1})$$

by splitting the complete state at the scanned symbol and writing the right-hand portion backwards. The reason for doing this is that one step of computation now changes only the right-hand ends of the elements of the pair, and these changes are easy to express arithmetically when we interpret the symbols $S_0 = \Box$, $S_1, \ldots, S_m$, q_0, $q_1, \ldots, q_n$ as digits in base b notation, where $b = m + n + 2$, and view the complete state pair as a pair of numbers. Before doing this, observe how blank tape is produced from nothing when we let $\Box$ be the zero in base b notation: the complete states

$$S_{k_u} \ldots S_{k_2}S_{k_1}q_iS_{j_1}S_{j_2} \ldots S_{j_v}$$

and

$$\Box\Box \ldots \Box S_{k_u} \ldots S_{k_2}S_{k_1}q_iS_{j_1}S_{j_2} \ldots S_{j_v}\Box\Box \ldots \Box,$$

whatever the number of $\Box$'s on either side, are both represented by the number pair $(S_{k_u} \ldots S_{k_2}S_{k_1}q_i, S_{j_v} \ldots S_{j_2}S_{j_1})$.

The quintuple $q_iS_jS_{j'}Lq_{i'}$ transforms the complete state

$$S_{k_u} \ldots S_{k_2}S_{k_1}q_iS_jS_{j_2} \ldots S_{j_v}$$

into

$$S_{k_u} \ldots S_{k_2}q_{i'}S_{k_1}S_{j'}S_{j_2} \ldots S_{j_v}$$

and hence transforms the complete state pair

$$(S_{k_u} \ldots S_{k_2}S_{k_1}q_i, S_{j_v} \ldots S_{j_2}S_j)$$

into

$$(S_{k_u} \ldots S_{k_2}q_{i'}, S_{j_v} \ldots S_{j_2}S_{j'}S_{k_1}).$$

More concisely, we can say that $q_iS_jS_{j'}Lq_{i'}$ transforms $(US_{k_1}q_i, VS_j)$ into $(Uq_{i'}, VS_{j'}S_{k_1})$ for any natural numbers U, V and any $S_{k_1} < b$. Thus the single leftmoving quintuple $q_iS_jS_{j'}Lq_{i'}$ corresponds to the b different leftmoving transformations, or *l-transformations* for short:

$$(US_{k_1}q_i, VS_j) \quad \text{goes to} \quad (Uq_{i'}, VS_{j'}S_{k_1})$$

for the b different values of S_{k_1}. Similarly, the rightmoving quintuple $q_i S_j S_{j'} R q_{i'}$ corresponds to b different *r-transformations*

$$(Uq_i, VS_{j_2}S_j) \quad \text{goes to} \quad (US_{j'}q_{i'}, VS_{j_2})$$

for the b different values of S_{j_2}. (Actually S_{j_2} could be absorbed into V, reducing these b transformations to one. We leave it as it is to maintain similarity with the l-transformations, and for greater symmetry in what follows.)

Invoking our interpretation of words as numerals, we can write the l-transformations in the form

$$(b^2U + A_l, bV + B_l) \quad \text{goes to} \quad (bU + C_l, b^2V + D_l) \ldots (l)$$

where $A_l = S_{k_1}q_i$, $B_l = S_j$, $C_l = q_{i'}$, $D_l = S_{j'}S_{k_1}$, and the r-transformations similarly in the form

$$(bU + A_r, b^2V + B_r) \quad \text{goes to} \quad (b^2U + C_r, bV + D_r) \ldots (r).$$

When convenient, we shall identify a machine M with the corresponding set of l- and r-transformations, taking $\{l\}$ and $\{r\}$ as disjoint sets of indices. Later we shall see that these transformations are very easy to simulate in finitely presented groups.

7. Unsolvability. To contemplate using a Turing machine to answer questions about other machines we first need a fixed finite alphabet in which to write a description, $\ulcorner M \urcorner$, of each machine M. This is because a Turing machine S can respond to only finitely many symbols, hence it cannot even read a question about an arbitrary M unless the symbols of the question are mentioned in its quintuples.

Without loss of generality we can assume that all machines work on finite subsets of the alphabet $\{\square, 1, 1', 1'', \ldots\}$, so that if we denote states by $q, q', q'', \ldots$ all quintuples can be written in the alphabet $\{q, q', q'', \ldots, \square, 1, 1', \ldots, R, L\}$, which we can reduce to the finite alphabet $\{q, \square, 1, {}', R, L\}$ by viewing $'$ as a whole, rather than part, symbol.

A machine is unambiguously described by concatenating its quintuples into a single word, e.g.

$$\left.\begin{matrix} q11Rq \\ q\square 1Rq' \end{matrix}\right\} \text{ by } q11Rqq\square 1Rq',$$

and finally we can code back into the agreed machine alphabet by

$$\begin{gathered} \square \leftrightarrow \square \\ 1 \leftrightarrow 1 \\ {}' \leftrightarrow 1' \\ q \leftrightarrow 1'' \\ R \leftrightarrow 1''' \\ L \leftrightarrow 1''''. \end{gathered}$$

The result of applying this process to the quintuples of a machine M will be called the *standard description*, $\ulcorner M \urcorner$, of M.

We consider the type of problem about machines M which consists of questions Q_M with a single parameter M. A machine S which receives input $\ulcorner M \urcorner$ may then be deemed to have received question Q_M, since any reasonable way of writing Q_M will be computable from $\ulcorner M \urcorner$. Likewise, there is no loss of generality in assuming that S answers "YES" by halting on 1, "NO" by halting on □. We therefore define a problem $\{Q_M\}$ to be *solvable* if there is an S which answers each question Q_M (in the above sense) correctly.

An unsolvable problem is now easily obtained by a little self-referential mischief; we ask the questions

Q_M: Does M eventually halt on □ after being given input $\ulcorner M \urcorner$?

Suppose there is a machine S which correctly answers Q_M for any value of M. When $M = S$ we get a contradiction because S interprets input $\ulcorner S \urcorner$ as the question Q_S and answers "YES" by halting on 1, in which case the true answer to Q_S is "NO"; S answers "NO" by halting on □, in which case the true answer to Q_S is "YES".

Thus we have proved the unsolvability of the

Special halting problem. For each Turing Machine M, decide whether M eventually halts on □ after receiving input $\ulcorner M \urcorner$.

The unsolvability of a similar halting problem was first proved by Turing [**1936**], by associating real numbers with certain machines and applying Cantor's diagonal argument. The related "self-referential" argument, which of course was used by Gödel [**1931**] in a different context, seems first to have been applied to Turing machines by Hermes [**1961**].

The unsolvability of the special halting problem is strong enough to prove the unsolvability of the isomorphism problem, and the word problem in a general form: for any group G and word w, decide whether $w = 1$ in G. However, if one wants a *specific* G with unsolvable word problem, a specific Turing machine with unsolvable halting problem is needed. This is the *universal Turing machine* of Turing [**1936**].

8. Universal Turing machines. When one reflects on standard descriptions, and how one might reconstruct the computation of a machine M on input I from $\ulcorner M \urcorner$ and a similar encoding $\ulcorner I \urcorner$ of I, it is plausible that a Turing machine T could do the work and hence simulate any machine M. We call such a T a *universal machine*. T starts on input $\ulcorner M \urcorner \ \ulcorner I \urcorner$ and then updates the encoding $\ulcorner I' \urcorner$ of M's tape expression by moving back and forth between $\ulcorner M \urcorner$ and $\ulcorner I' \urcorner$, keeping track of the currently active quintuple in $\ulcorner M \urcorner$ and the current scanned symbol in $\ulcorner I' \urcorner$ by suitable marks. The energetic reader is urged to fill in the details of this idea, since it is probably easier to construct a universal machine for oneself than to read any of the accounts in the literature. The less energetic reader may appeal to Church's thesis: the simulation of M on I is clearly computable, hence there is a Turing machine that does it.

The coding into machine alphabet used in the previous section has the convenient property that □ encodes itself, hence M eventually halts on □ after receiving input $\ulcorner M \urcorner$ if and only if T eventually halts on □ after receiving input $\ulcorner M \urcorner \ \ulcorner \ulcorner M \urcorner \urcorner$. Thus from the unsolvability of the special halting problem we obtain unsolvability of the halting problem for T on special inputs $\ulcorner M \urcorner \; \ulcorner \ulcorner M \urcorner \urcorner$, and *a fortiori* the unsolvability of the

Halting problem for T. For each input J, decide whether T eventually halts on $\square$ after receiving input J.

This problem can obviously be rephrased as an unsolvable problem about the l- and r-transformations used in §6. However, a more convenient problem is obtained if we first modify T to a machine T' which imitates T until T halts on $\square$, if ever, then erases the whole tape and halts in a previously unused state q_0. T' is easily constructed by adding end markers to the alphabet of T and using them to enclose the marked portion of tape at all times (pushing them further out whenever more space is needed). When the time comes to erase the tape, T' first moves to the right end marker, then erases everything back to and including the left end marker, before halting by going into state q_0.

Then if we identify T' with the corresponding set of l- and r-transformations, and if (U_I, V_I) represents T' starting on input I we have

$$T \text{ eventually halts on } \square \text{ after receiving input } I$$
$$\Leftrightarrow (U_I, V_I) \text{ is convertible to } (q_0, 0) \text{ by } T'.$$

Since the former problem is unsolvable, so is the latter.

Finally we construct a system T^* by adding transformations which allow $(q_0, 0)$ to be converted to the simplest possible pair, $(0, 0)$. We make these transformations of the l-type in §6, i.e.

$$\left(b^2U + A_l, bV + B_l\right) \quad \text{goes to} \quad \left(bU + C_l, b^2V + D_l\right)$$

by setting $A_l = S_{k_1}q_0$, $B_l = 0$, $C_l = S_{k_1}$, $D_l = 0$ for the b possible l values corresponding to different S_{k_1}. These certainly convert $(q_0, 0)$ to $(0, 0)$, but more generally they can only be applied to a complete state pair of the form $(S_{k_u} \ldots S_{k_1}q_0, S_{j_r} \ldots S_{j_1}\square)$, which represents the complete state $S_{k_u} \ldots S_{k_1}q_0\square S_{j_1} \ldots S_{j_r}$, and this *cannot arise* until the tape is empty, i.e. when the complete state is $q_0\square$, because q_0 is not used until this stage.

Combining the results of the last three sections we have

THEOREM 1. *There is a finite set T^* of transformations of pairs of natural numbers, of the forms*

$$\left(b^2U + A_l, bV + B_l\right) \quad \text{goes to} \quad \left(bU + C_l, b^2V + D_l\right),$$
$$\left(bU + A_r, b^2V + B_r\right) \quad \text{goes to} \quad \left(b^2U + C_r, bV + D_r\right)$$

for which the problem of deciding, for each pair (U_I, V_I), whether (U_I, V_I) is convertible to $(0, 0)$ *by T^*, is unsolvable.*

9. Semigroups. The first success of computability theory outside logic was in the theory of semigroups. Post [**1946, 1947**] proved the unsolvability of two simple semigroup problems, though he did not state them as such. The 1946 problem is generally known as the *Post correspondence problem*, and it can be stated as follows.

Let S_n be the free semigroup on generators $a_1, \ldots, a_n$ and let $S_n \times S_n$ be the direct product. For any finite set of pairs $(A_i, B_i) \in S_n \times S_n$, let $S_{\{(A_i, B_i)\}}$

be the subsemigroup they generate, and let D denote the subsemigroup generated by $(a_1, a_1), \ldots, (a_n, a_n)$. Then the problem is to decide whether $S_{\{(A_i,B_i)\}} \cap D = \varnothing$.

Post's proof was the belated fruit of 20 years' work on combinatorial definitions of computability, in which he had anticipated the main results of Gödel and Church, but in a form he considered unpublishable, and arrived at the Turing machine concept at the same time as Turing (see the 1941 paper of Post in Davis [**1965**]). With the correspondence problem he finally moved ahead of the field. Church then suggested that he tackle the word problem for semigroups, which can be stated as follows.

Given a finite set of equations $P_i = Q_i$ between words on an alphabet $\{a_1, \ldots, a_n\}$, and any words X, Y, decide whether $X = Y$ is a consequence of the equations. (The notion of consequence of course differs from that defined for groups in §1 in the absence of equations $a_i a_i^{-1} = a_i^{-1} a_i = 1$.)

Post [**1947**] proved the word problem unsolvable by using words to represent complete machine states, much as we have done in §6, and using equations to make the changes which occur with successive steps of computation. In order to have a specific word represent the halt-on-□ situations, he introduces a new symbol q which is created from $q_i\square$ whenever the machine has no response to this situation. Additional equations of the forms $qs = q$, $sq = q$, for all other symbols s, enable q to "eat up" the rest of the word. Then the halting problem is equivalent to deciding, for a given word w (representing the initial complete state), whether $w = q$ in the semigroup. By writing down the equations which reflect computation by the universal machine T, one obtains a fixed, finitely presented semigroup with unsolvable word problem.

A similar proof was obtained independently by Markov [**1947**].

Unfortunately, Post's semigroup cannot be embedded in a group. This is obvious from the equations $sq = q$, which in a group would imply $s = 1$, and hence collapse everything to free generators q_i. Thus the word problem for groups remained open.

Turing first heard about the word problem for groups in the late 1940's (see Crossley [**1974**, p. 54]), became absorbed in it, and after about 10 days' work announced a proof that it was unsolvable. By the time he was due to deliver his proof in a seminar, he had discovered a mistake, but he still managed to prove unsolvability for semigroups with cancellation. In a semigroup with cancellation one can derive $A = B$ from $XA = XB$ or $AX = BX$ for any word X, though this still does not imply that the semigroup can be embedded in a group. Nevertheless, Turing's result (Turing [**1950**]) was a big advance, and some of his ideas were crucial in the later proofs for groups.

For example, Turing prevents the semigroup from collapsing under cancellation by the use of *record symbols*. Instead of replacing the part P_i of the complete state by Q_i, one replaces it by $\sigma_i Q_i \tau_i$, where σ_i, τ_i are symbols which "record" the fact that the ith equation is being used. This protects q from being cancelled in case $P_i = sq$, $Q_i = q$, though of course new relations are needed to ultimately remove the record symbols, and this creates other problems. Turing could overcome these problems for cancellation semigroups, but not for groups.

10. Groups. Just as the word problem for semigroups followed the Post correspondence problem, the word problem for groups was not settled until unsolvability had first been proved for something more complicated. In 1951 Boone proved the unsolvability of what he called the *quasi-Magnus problem* for a finitely presented group G: to decide, given a subset $\{a_1, \ldots, a_e\}$ of the generators of G and any word w, whether w is equal to a word on $\{a_1, \ldots, a_e\}$ with positive exponents. (The corresponding problem for arbitrary exponents, in other words, deciding whether w is in the *subgroup* generated by $\{a_1, \ldots, a_e\}$, is known as the *Magnus problem*. It was solved for one-relator groups by Magnus [**1932**].)

A Magnus-type problem arises naturally when one attempts to simulate a Turing machine by equations in a group, using record symbols to prevent unwanted cancellations. Record symbols accumulate as the computation proceeds, so a computation leading to halt-on-□ cannot be identified by a specific final word, but rather a specific subword in.an unpredictable mess of record symbols. Thus the question equivalent to the halting problem is whether a given word equals a word of a certain form on a certain subset of the generators. Boone's result was contained in his 1952 Princeton dissertation, supervised by Church, and published in Boone [**1954a,b, 1955a,b**]. In 1956 Boone succeeded in modifying his construction so that record symbols could be removed at the end of the computation, thus obtaining a proof of the unsolvability of the word problem.

In the meantime, Novikov had published his unsolvability proof in 1955 (though Boone's work was independent of this). Novikov takes Turing's semigroup with cancellation as his starting point, but he and Boone use similar devices for controlling the movement of symbols. For example, both use commuting relations $kS_i = S_ik$ which allow k's to "filter through" words on $\{S_i\}$, and quasi-commuting relations $xS_i = S_ix^2$ which allow unlimited rightward movement of x's through words on $\{S_i\}$, but restrict leftward movement, since each time a power of x moves left across an S_i, its exponent is halved. It turns out, ultimately, that *only* these types of relation are needed, together with Turing-style use of record symbols (see §§13 and 14).

Novikov and Boone proved all results about relations in their groups by long combinatorial arguments from first principles. Little by little, people noticed that their combinatorial lemmas could be proved by known arguments from group theory, using in particular the free product with amalgamation of Schreier [**1927**], and the "HNN" construction of Higman, B. H. Neumann, H. Neumann [**1949**]. Boone [**1959**] reports receiving suggestions along these lines from R. H. Fox in [**1955**] and Higman in [**1957**]. In [**1958**], Britton (who was a student of B. H. Neumann) was the first to actually use these methods in an unsolvability proof for the word problem, extending them to a special case of what is now known as Britton's lemma.

The full Britton's lemma appears in Britton [**1963**], where it is used to completely replace the combinatorial arguments of Boone [**1959**], giving the shortest proof then known. Later it was shown that Novikov's arguments could be similarly replaced. It was astonishing to find a single group-theoretic explanation of so many combinatorial facts, and this discovery undoubtedly helped to put the HNN construction on the map.

11. The HNN construction and normal forms. Given a group G, and pairs of elements b_i, c_i, suppose that $b_i \mapsto c_i$ defines an isomorphism of the subgroups B, C of G generated by $\{b_i\}$, $\{c_i\}$ respectively. Then the group H obtained from G by adding the relations $t^{-1}b_it = c_i$, which we shall write

$$H = G \cup \langle t;\ \{t^{-1}b_it = c_i\}\rangle,$$

is called an *HNN extension of G with stable letter t*. Notice that $t^{-1}b_{i_1}^{\varepsilon_1} \ldots b_{i_k}^{\varepsilon_k}t = c_{i_1}^{\varepsilon_1} \ldots c_{i_k}^{\varepsilon_k}$, where $b_{i_1}^{\varepsilon_1} \ldots b_{i_k}^{\varepsilon_k} = b$ is an arbitrary element of B, so if ϕ: $B \to C$ denotes the isomorphism determined by $b_i \mapsto c_i$ we can also write

$$H = G \cup \langle t;\ \{t^{-1}bt = \phi(b)|b \in B\}\rangle.$$

Higman, B. H. Neumann, H. Neumann **[1949]** proved that G embeds in H, which we abbreviate $G \hookrightarrow H$. More precisely, adding the relations $t^{-1}bt = \phi(b)$ does not yield any new consequence relations among the generators of G. Adding new relations without affecting the old is of course exactly what we want to do in constructing groups to simulate machines.

Actually, a stronger result is needed for applications to the word problem, and to motivate it we consider a normal form for elements in an HNN extension.

A typical word in $H = G \cup \langle t; \{t^{-1}bt = \phi(b)|b \in B\}\rangle$ looks like

$$g_0t^{\varepsilon_1}g_1t^{\varepsilon_2} \ldots t^{\varepsilon_k}g_k$$

where each $\varepsilon_i = \pm 1$ and each g_i is a word in the generators of G (possibly 1). Each g_i can be factored into an element of B or C and a "residue", i.e. a coset representative of G mod B or C. Writing the relation $t^{-1}bt = \phi(b)$ as $t\phi(b) = bt$ or $t^{-1}b = \phi(b)t^{-1}$, we see that an element $\phi(b) \in C$ can always pass to the left across t, becoming b on the other side, while $b \in B$ can always pass to the left across t^{-1}, becoming $\phi(b)$ on the other side. This suggests normalizing the word by draining off elements of B or C to the left, leaving residues stuck between the t's.

To make this process precise (though not necessarily computable), we chose specific coset representatives; g^B of the coset Bg, g^C of the coset Cg, with 1 as the representative of both B and C. Then we work from right to left as follows.

If $\varepsilon_k = -1$ we factorize g_k into $B_kg_k^B$, where $B_k \in B$, so

$$t^{\varepsilon_k}g_k = t^{-1}B_kg_k^B = \phi(B_k)t^{-1}g_k^B = \phi(B_k)t^{\varepsilon_k}g_k^B.$$

Similarly, if $\varepsilon_k = +1$ we factorize g_k into $C_kg_k^C$, where $C_k \in C$, and

$$t^{\varepsilon_k}g_k = tC_kg_k^C = \phi^{-1}(C_k)tg_k^C = \phi^{-1}(C_k)t^{\varepsilon_k}g_k^C.$$

(Since ϕ is an isomorphism, ϕ^{-1} is well defined.) We now have either $g_{k-1}\phi(B_k)$ or $g_{k-1}\phi^{-1}(C_k)$ between $t^{\varepsilon_{k-1}}$ and t^{ε_k}. We factorize it similarly, according to the sign of ε_{k-1}, and continue passing elements of B or C to the left, leaving coset representatives behind. If at any stage t and t^{-1} appear with only 1 between them, they are cancelled. The final result is a word of the form

$$g_0't^{\delta_1}g_1't^{\delta_2} \ldots t^{\delta_e}g_e', \qquad \delta_i = \pm 1$$

where (i) g'_0 is an arbitrary element of G,

(ii) $\delta_i = -1 \Rightarrow g'_i$ is a coset representative of G mod B,

(iii) $\delta_i = +1 \Rightarrow g'_i$ is a coset representative of G mod C,

(iv) t, t^{-1} do not occur as consecutive letters.

This word is still not unique, because g'_i can be any word from the equivalence class $[g'_i]$ in G. Let us denote the class of words which result from replacing g'_i by other representatives of $[g'_i]$ by

$$[g'_0]t^{\delta_1}[g'_1]t^{\delta_2}\dots t^{\delta_e}[g'_e]$$

and call this class a *normal form* of the element of H it represents.

12. Britton's lemma. This follows from

THEOREM 2. *The normal form of an element of H is unique.*

PROOF. We shall faithfully represent H as a group of permutations of the set N of normal forms. For each $h \in H$ we shall define a mapping Φ_h: $N \to N$ with the properties

(i) Φ_1 is the identity,

(ii) $\Phi_{h_1}\Phi_{h_2} = \Phi_{h_1h_2}$,

(iii) $\Phi_{\text{normal form } h}(1) = \text{normal form } h$.

From (i) and (ii) it follows that $h \mapsto \Phi_h$ is a homomorphism, in particular $\Phi_h\Phi_{h^{-1}} = \text{identity}$, so Φ_h is invertible, hence a permutation. From (iii) it follows that normal forms are unique, because Φ_{h_1}, Φ_{h_2} for different normal forms h_1, h_2 send 1 to different places, so Φ_{h_1}, Φ_{h_2} are different permutations and hence represent different elements h_1, h_2.

The mapping Φ_h is defined to be "multiply on the left and reduce to normal form, one letter of h at a time". Then (i)–(iii) are clear, but since the word for an element $h \in H$ is not unique, the problem is to show that Φ_h is well defined. This requires showing that equivalent words determine the same permutation, in other words, that each defining relator of H determines the identity permutation. For a relator g of G this is clear, because

$$\Phi_g([g'_0]t^{\delta_1}\dots t^{\delta_e}[g'_e]) = [gg'_0]t^{\delta_1}\dots t^{\delta_e}[g'_e] = [g'_0]t^{\delta_1}\dots t^{\delta_e}[g'_e]$$

since $g = 1$ in G. (Although we are supposed to apply g one letter at a time, it is clear that these letters just accumulate to the left of g'_0, since no interaction with t is possible. We shall similarly present just the end results of letter-by-letter accumulations below.)

The relators involving t are tt^{-1}, $t^{-1}t$ and $t^{-1}bt\phi(b)^{-1}$. We check just the last of these, since the first two are similar, but easier.

$$\begin{aligned}\Phi_{t^{-1}bt\phi(b)^{-1}}([g'_0]t^{\delta_1}[g'_1]t^{\delta_2}\dots) &= \Phi_{t^{-1}}\Phi_b\Phi_t\Phi_{\phi(b)^{-1}}([g'_0]t^{\delta_1}[g'_1]t^{\delta_2}\dots)\\ &= \Phi_{t^{-1}}\Phi_b\Phi_t([\phi(b)^{-1}g'_0]t^{\delta_1}[g'_1]t^{\delta_2}\dots)\\ &= \Phi_{t^{-1}}\Phi_b\,(\text{normal form of } t[\phi(b)^{-1}g'_0]t^{\delta_1}[g'_1]t^{\delta_2}\dots).\end{aligned}$$

There are now three cases to consider:
(a) $[\phi(b)^{-1}g_0'] = 1$, i.e. $\phi(b) = g_0'$, and $\delta_1 = -1$;
(b) $[\phi(b)^{-1}g_0'] = 1$, i.e. $\phi(b) = g_0'$, and $\delta_1 = +1$;
(c) $[\phi(b)^{-1}g_0'] \neq 1$.
In case (a), t and t^{δ_1} cancel and we continue with

$$\begin{aligned}
\Phi_{t^{-1}}\Phi_b([g_1']t^{\delta_2}\dots) &= \Phi_{t^{-1}}([bg_1']t^{\delta_2}\dots)\\
&= \text{normal form of } t^{-1}[bg_1']t^{\delta_2}\dots\\
&= [\phi(b)]t^{-1}[g_1']t^{\delta_2}\dots \quad \text{since } g_1' \text{ is a coset representative}\\
&= [g_0']t^{\delta_1}[g_1']t^{\delta_2}\dots \quad \text{since } \phi(b) = g_0',\ \delta_1 = -1 \text{ by hypothesis.}
\end{aligned}$$

In case (b) we continue with

$$\begin{aligned}
\Phi_{t^{-1}}\Phi_b(t[1]t[g_1']t^{\delta_2}\dots) &= \Phi_{t^{-1}}([b]t[1]t[g_1']t^{\delta_2}\dots)\\
&= \text{normal form of } t^{-1}[b]t[1]t[g_1']t^{\delta_2}\dots\\
&= \text{normal form of } [\phi(b)]t^{-1}t[1]t[g_1']t^{\delta_2}\dots\\
&= [\phi(b)]t[g_1']t^{\delta_2}\dots, \quad \text{cancelling } t^{-1}t\\
&= [g_0']t^{\delta_1}[g_1']t^{\delta_2}\dots \quad \text{since } \phi(b) = g_0',\ \delta_1 = 1 \text{ by hypothesis.}
\end{aligned}$$

In case (c) we let $\phi(b)^{-1}g_0' = \phi(\bar{b}_0)^{-1}g_0^c$, where $\phi(\bar{b}_0)^{-1} \in C$ and g_0^c is the coset representative, and continue with

$$\begin{aligned}
&\Phi_{t^{-1}}\Phi_b\Big(\text{normal form of } t\Big[\phi(\bar{b}_0)^{-1}g_0^c\Big]t^{\delta_1}\dots\Big)\\
&= \Phi_{t^{-1}}\Phi_b\big(\big[\bar{b}_0^{-1}\big]t[g_0^c]t^{\delta_1}\dots\big)\\
&= \Phi_{t^{-1}}\big(\big[b\bar{b}_0^{-1}\big]t[g_0^c]t^{\delta_1}\dots\big)\\
&= \text{normal form of } t^{-1}\big[b\bar{b}_0^{-1}\big]t[g_0^c]t^{\delta_1}\dots\\
&= \text{normal form of } \big[\phi\big(b\bar{b}_0^{-1}\big)\big]t^{-1}t[g_0^c]t^{\delta_1}\dots \quad \text{since } b\bar{b}_0^{-1} \in B\\
&= \big[\phi\big(b\bar{b}_0^{-1}\big)g_0^c\big]t^{\delta_1}\dots, \quad \text{cancelling } t^{-1}t\\
&= \Big[\phi(b)\phi\big(\bar{b}_0\big)^{-1}g_0^c\Big]t^{\delta_1}\dots \quad \text{since } \phi \text{ is a homomorphism}\\
&= [g_0']t^{\delta_1}\dots \quad \text{since } \phi\big(\bar{b}_0\big)^{-1}g_0^c = \phi(b)^{-1}g_0' \text{ by definition.}
\end{aligned}$$

Thus $\Phi_{t^{-1}bt\phi(b)^{-1}}$ is indeed the identity. □

Corollary 1 (Higman, B. H. Neumann, H. Neumann **[1949]**). *$G \hookrightarrow H$.*

Proof. Immediate, since a $[g] \in G$ is identical with its normal form.

Corollary 2 (Britton's lemma, Britton **[1963]**). *If w is a word involving t and $w = 1$, then w contains either a subword $t^{-1}bt$, where $b \in B$, or a subword tct^{-1}, where $c \in C$.*

PROOF. The normal form of w is 1, hence t's must be cancelled in the normalization process. Suppose for example that $t^{-1}bt$ is a subword of w whose t's get cancelled in the normalization process. Normalization inserts a word $\bar{b} \in B$ between b and t, and the t's cancel only if $b\bar{b} \in B$, because only then can $b\bar{b}$ be moved to the left leaving no residue. But $b\bar{b} \in B \Leftrightarrow b \in B$.

Similarly, if tct^{-1} is a subword of w whose t's are cancelled during normalization, then $c \in C$. □

COROLLARY 3 (GENERALIZED BRITTON'S LEMMA). *If*

$$H_1 = G \cup \langle t_1;\ \{t_1^{-1}\, dt_1 = \phi_1(d) | d \in D_1\}\rangle$$
$$\vdots$$
$$H_n = H_{n-1} \cup \langle t_n;\ \{t_n^{-1}\, dt_n = \phi_n(d) | d \in D_n\}\rangle$$

are successive HNN extensions, and if w is a word involving t_i's which equals 1 in H_n, then w contains a subword $t_i^{-1}d_it_i$ where $d_i \in D_i$, or a subword $t_ie_it_i^{-1}$, where $e_i \in E_i = \phi_i(D_i)$.

PROOF. Consider the t_i of highest index in w, then view w as a word in H_i. Since H_i is an HNN extension of H_{i-1} with stable letter t_i, the conclusion is immediate by Corollary 2. □

The embedding $G \hookrightarrow H$ and Britton's lemma were first proved using the theory of free products with amalgamation, based on a similar technique of permuting normal forms developed by Van der Waerden **[1948]** and B. H. Neumann **[1954]**. The latter paper also contains a direct proof of $G \hookrightarrow H$, attributed to Philip Hall, which uses permutations and coset representatives, but not normal forms. The direct application of permutations and normal forms to HNN theory is due to Schupp **[1974]**.

13. A group with unsolvable Magnus problem. We begin with the free group of rank 3

$$F = \langle x, y, z; - \rangle$$

and encode natural number pairs (i, j) by the elements

$$p(i, j) = x^izy^j.$$

We want to reflect the l- and r-transformations, used to represent Turing machines in §6, by isomorphisms ϕ_l, ϕ_r in F which act appropriately on the corresponding $p(i, j)$. Since an l-transformation sends

$$(b^2U + A_l, bV + B_l) \quad \text{to} \quad (bU + C_l, b^2V + D_l)$$

we want ϕ_l to send

$$x^{b^2U+A_l}zy^{bV+B_l} \quad \text{to} \quad x^{bU+C_l}zy^{b^2V+D_l}.$$

This can be done if ϕ_l sends x^{b^2U} to x^{bU}, $x^{A_l}zy^{B_l}$ to $x^{C_l}zy^{D_l}$, y^{bV} to y^{b^2V}, which in turn can be accomplished by the map

$$\phi_l\colon x^{b^2} \mapsto x^b, \qquad x^{A_l}zy^{B_l} \mapsto x^{C_l}zy^{D_l}, \qquad y^b \mapsto y^{b^2}.$$

This map is indeed an isomorphism, since we saw in §4 that $\{x^{b^2}, x^{A_l}zy^{B_l}, y^b\}$ and $\{x^b, x^{C_l}zy^{D_l}, y^{b^2}\}$ are bases for free subgroups of rank 3 in F.

Thus ϕ_l: $x^{b^2} \mapsto x^b, p(A_l, B_l) \mapsto p(C_l, D_l), y^b \mapsto y^{b^2}$ is an isomorphism in F which sends $p(b^2U + A_l, bV + B_l)$ to $p(bU + C_l, b^2V + D_l)$. Similarly, ϕ_r: $x^b \mapsto x^{b^2}$, $p(A_r, B_r) \mapsto p(C_r, D_r)$, $y^{b^2} \mapsto y^b$ is an isomorphism in F which sends $p(bU + A_r, b^2V + B_r)$ to $p(b^2U + C_r, bV + D_r)$.

We can now extend F to a group $H(T^*)$ in which ϕ_l is induced by conjugation with stable letter t_l, and ϕ_r is induced by conjugation with stable letter t_r, where $\{l\}$ and $\{r\}$ index the transformations making up T^* in §8. Namely, let $H(T^*)$ be the result

$$F \cup \langle \{t_l\}, \{t_r\}; \{t_l^{-1}x^{b^2}t_l = x^b\}, \{t_l^{-1}p(A_l, B_l)t_l = p(C_l, D_l)\}, \{t_l^{-1}y^bt_l = y^{b^2}\},$$
$$\{t_r^{-1}x^bt_r = x^{b^2}\}, \{t_r^{-1}p(A_r, B_r)t_r = p(C_r, D_r)\}, \{t_r^{-1}y^{b^2}t_r = y^b\}\rangle$$

of a series of HNN extensions with stable letters t_l, t_r.

It is clear that if (U, V), (U', V'), (U'', V''), ... is a sequence of complete state pairs which result from successive steps of computation by T^*, then $p(U', V')$, $p(U'', V'')$, ... can be produced from $p(U, V)$ by ϕ_l and ϕ_r isomorphisms, and hence by conjugations with t_l's and t_r's. To this extent $H(T^*)$ reflects computation in T^* (with the t_l's and t_r's serving as "record symbols"), but we have to show that $H(T^*)$ does not also allow "fake" computations which have no counterpart in T^*. The following lemmas will guarantee this.

LEMMA 1. *If (X, Y) is a complete state pair, then at most one ϕ_l or ϕ_r applies to $p(X, Y)$.*

PROOF. If ϕ_l applies to $p(X, Y)$ then $p(X, Y)$ must be in the domain $\langle x^{b^2}, x^{A_l}zy^{B_l}, y^b; -\rangle$ of ϕ_l, i.e. $x^Xzy^Y \in \langle x^{b^2}, x^{A_l}zy^{B_l}, y^b; -\rangle$. Then by Proposition (ii) of §4 we have

$$X = A_l + m_1b^2, \qquad Y = B_l + n_1b \quad \text{for some integers } m_1, n_1.$$

But since A_l is a two-digit number in base b, and B_l a single digit (see §6), these equations uniquely determine A_l and B_l, and hence ϕ_l, since there is at most one l-transformation for given values of A_l, B_l.

There is a similar proof for ϕ_r. Finally, it is impossible for both a ϕ_l and a ϕ_r to apply to $p(X, Y)$, as this would require

$$X = A_l + m_1b^2 = A_r + m_2b,$$
$$Y = B_l + n_1b = B_r + n_2b^2 \quad \text{for some integers } m_2, n_2,$$

hence

$$\text{last digit of } A_l = q_i = \text{last digit of } A_r = q_i,$$
$$\text{last digit of } B_l = S_j = \text{last digit of } B_r = S_j,$$

whereas there is at most one quintuple beginning with q_iS_j. □

The encoding of complete states by numbers makes Lemma 1 possible. If one uses instead a more obvious representation of the complete state by a word on $\{q_i\}$, $\{S_j\}$, like that in §6, and works in the group with generators

$\{q_i\}$, $\{S_j\}$ (as Boone [1959] and Britton [1963] do), then one needs isomorphisms which send complete states to their successors. For example, with the quintuple $q_i S_{j_1} S_{j'_1} R q_{i'}$, one wants the subword $q_i S_{j_1} S_{j_2}$ of the complete state replaced by $S_{j'_1} q_{i'} S_{j_2}$, while the rest of the word is preserved. Thus we need

$$S_j \mapsto S_j \quad \text{for each } j, \quad q_i S_{j_1} S_{j_2} \mapsto S_{j'_1} q_{i'} S_{j_2}.$$

This is indeed an isomorphism, between the free groups with bases $\{S_j\}$, $q_i S_{j_1} S_{j_2}$ and $\{S_j\}$, $S_{j'_1} q_{i'} S_{j_2}$, but we cannot guarantee that it is the only isomorphism which applies to the word. The free group with basis $\{S_j\}$, $q_i S_{j_1} S_{j_2}$ also has basis $\{S_j\}$, q_i, hence any isomorphism corresponding to a quintuple which begins with q_i will also apply to it.

To overcome this problem, Boone introduces some remarkable but complicated modifications, for example a new generator x and relations $xS_j = S_j x^2$ which Britton views as an isomorphism $x \mapsto x^2$ being induced by conjugation with stable letter S_j. In the Cohen and Aanderaa proof such isomorphisms are miraculously merged with the isomorphism $p(A, B) \mapsto p(C, D)$ which effects the change in the neighbourhood of the scanned symbol by dissolving the q_i's and S_j's altogether, encoding their sequence by the exponents of x and y. (Actually Cohen and Aanderaa use a more complicated encoding than ours, which does not permit them to work in a free group, however the idea is the same.)

A second possible source of fake computations in $H(T^*)$ is the fact that an equation $P = Q$ is, so to speak, *bidirectional*. It allows subword P to be replaced by Q, or subword Q to be replaced by P, whereas only one of these directions has a computational interpretation, the other corresponding to a fake, backward, computational step. This problem already arises with semigroups, and was solved in that context by Post [1947]. To use Post's argument in our present context we introduce the *halting subgroup* $H_0(T^*) \subset F \subset H(T^*)$ by

$$H_0(T^*) = \langle \{p(U, V) | (U, V) \text{ is a complete state pair which } T^* \text{ converts to } (0, 0)\}\rangle.$$

Here, and below, we use the notation $\langle S \rangle$, $\langle x_1, x_2, \ldots \rangle$ or similar to denote the subgroup of $H(T^*)$ generated by the set S of elements $x_1, x_2, \ldots$.

LEMMA 2. *$H_0(T^*)$ is closed under $\phi_l^{\pm 1}$ and $\phi_r^{\pm 1}$.*

PROOF. Suppose $p(U, V) \in H_0(T^*)$. By Lemma 1 at most one $\phi_l p(U, V)$ or $\phi_r p(U, V) = p(U', V')$ is defined, and if so (U', V') is the result of the unique corresponding l- or r-transformation which applies to (U, V). Hence if T^* converts (U, V) to $(0, 0)$ it in fact converts (U', V') to $(0, 0)$ in the process.

Thus $p(U', V') \in H_0(T^*)$, so $H_0(T^*)$ is closed under ϕ_l and ϕ_r.

Closure under ϕ_l^{-1} and ϕ_r^{-1} is more trivial and does not depend on the determinacy of T^*. Namely, if

$$\phi_l^{-1} p(U, V) = p(\bar{U}, \bar{V})$$

we have

$$\phi_l p(\bar{U}, \bar{V}) = p(U, V),$$

so (U, V) results from $(\bar{U}, \bar{V})$ by an l-transformation. Then if T^* converts (U, V) to $(0, 0)$ it also converts $(\bar{U}, \bar{V})$ to $(0, 0)$, i.e. $p(U, V) \in H_0(T^*) \Rightarrow \phi_l^{-1}p(U, V) \in H_0(T^*)$. The argument is the same for ϕ_r^{-1}. □

An application of Britton's lemma now shows that Lemmas 1 and 2 dispose of all possibilities for fake computations. That is, no interaction of $\{t_l\}$ and $\{t_r\}$ with $H_0(T^*)$ can yield a $p(X, Y) \notin H_0(T^*)$ when (X, Y) is a complete state pair:

Lemma 3. $F \cap \langle H_0(T^*), \{t_l\}, \{t_r\}\rangle = H_0(T^*)$.

Proof. Suppose that w is a word in $\langle H_0(T^*), \{t_l\}, \{t_r\}\rangle$ which equals a word f on x, y, z. Then $wf^{-1} = 1$ in $H(T^*)$, so if w contains any t_i the generalized Britton's lemma says it contains a subword $t_i^{-1}d_it_i$ or $t_ie_it_i^{-1}$ where $d_i \in H_0(T^*) \cap$ domain ϕ_i, $e_i \in H_0(T^*) \cap$ range ϕ_i and $i \in \{l\} \cup \{r\}$. By Lemma 2 we then have $t_i^{-1}d_it_i = \phi_i(d_i) \in H_0(T^*)$, $t_ie_it_i^{-1} = \phi_i^{-1}(e_i) \in H_0(T^*)$, so these t_i's can be eliminated. Continuing in this way, we can eliminate all t_i's in w, showing that $w \in H_0(T^*)$. □

Lemma 3 is actually a special case of a result on subgroups of HNN extensions, proved by Higman [**1961**] without using Britton's lemma. However, Britton's lemma provides the most uniform explanation of this result and others we shall need in the next section.

Now we are ready for the main result of the present section:

Theorem 3. *The Magnus problem for $H(T^*)$ is unsolvable.*

Proof. It will suffice to show that, for a complete state pair (U, V),

$$T^* \text{ converts } (U, V) \text{ to } (0, 0) \Leftrightarrow p(U, V) \in \langle z, \{t_l\}, \{t_r\}\rangle,$$

since the left-hand side is unsolvable by Theorem 1, and the right-hand side is decided by a solution to the Magnus problem for $H(T^*)$.

($\Rightarrow$). If l- and r-transformations in T^* convert (U, V) to $(0, 0)$, then $p(U, V)$ is converted to $p(0, 0)$ by the corresponding isomorphisms ϕ_l and ϕ_r, and hence by a series of conjugations by t_l's and t_r's. That is,

$$W^{-1}p(U, V)W = p(0, 0) = z \quad \text{for some word } W \text{ on } \{t_l\}, \{t_r\}.$$

But then, $p(U, V) = WzW^{-1} \in \langle z, \{t_l\}, \{t_r\}\rangle$.

($\Leftarrow$). Suppose $p(U, V) \in \langle z, \{t_l\}, \{t_r\}\rangle$. Since $z = p(0, 0) \in H_0(T^*)$ we have, *a fortiori*, $p(U, V) \in \langle H_0(T^*), \{t_l\}, \{t_r\}\rangle$, and hence by Lemma 3, $p(U, V) \in H_0(T^*)$. Then, by definition of $H_0(T^*)$, T^* converts (U, V) to $(0, 0)$. □

14. Unsolvability of the word problem and isomorphism problem. It is now easy to construct a group with unsolvable word problem from $H(T^*)$ by a device of Boone [**1959**]. Namely, let

$$K(T^*) = H(T^*) \cup \langle k; k^{-1}zk = z, \{k^{-1}t_lk = t_l\}, \{k^{-1}t_rk = t_r\}\rangle.$$

Since $z \mapsto z$, $\{t_l \mapsto t_l\}$, $\{t_r \mapsto t_r\}$ is obviously an isomorphism in $H(T^*)$, $K(T^*)$ is an HNN extension of $H(T^*)$ with stable letter k.

THEOREM 4. *The word problem for $K(T^*)$ is unsolvable.*

PROOF. Since k commutes with z, $\{t_l\}$, $\{t_r\}$, a word $p(U, V) \in \langle z, \{t_l\}, \{t_r\}\rangle$ satisfies

$$kp(U, V) = p(U, V)k.$$

Conversely, suppose this equation holds, in other words that

$$kp(U, V)k^{-1}p(U, V)^{-1} = 1.$$

Since the only occurrences of k, k^{-1} on the left-hand side are those explicitly shown, Britton's lemma tells us immediately that $p(U, V) \in \langle z, \{t_l\}, \{t_r\}\rangle$.

Thus $p(U, V) \in \langle z, \{t_l\}, \{t_r\}\rangle \Leftrightarrow kp(U, V)k^{-1}p(U, V)^{-1} = 1$. A solution to the word problem would decide the right-hand side, hence the left-hand side, but this is impossible by the unsolvability of the Magnus problem in Theorem 3. □

The isomorphism problem was first proved unsolvable independently by Adian [**1957**] and Rabin [**1958**]. Actually these authors proved unsolvability of a large number of group-theoretic problems; the isomorphism problem is quite easily settled once one has a group with unsolvable word problem in which every element $\neq 1$ has infinite order, as was pointed out by Novikov [**1958**].

To see that every element $\neq 1$ in $K(T^*)$ has infinite order, observe that this is certainly true of the free group F we started with. It remains true under each successive HNN extension; elements not involving the new stable letter t retain their orders by the embedding of Corollary 1 to Theorem 2, while an element involving t is easily seen to be of infinite order by applying Britton's lemma to powers of its normal form.

THEOREM 5. *The isomorphism problem is unsolvable.*

PROOF. Take the group $K(T^*)$ with unsolvable word problem and no elements $\neq 1$ of finite order, and for each w in the generators of $K(T^*)$, which we rename $a_1, \ldots, a_p$, construct the group

$$K_w(T^*) = K(T^*) \cup \langle\{k_i\}; \{k_i^{-1}wk_i = a_i\}\rangle.$$

If $w = 1$ in $K(T^*)$ then each $a_i = 1$ in $K_w(T^*)$ and hence

$$K_w(T^*) = \langle\{k_i\}; -\rangle = \text{free group of rank } p.$$

If $w \neq 1$ in $K(T^*)$ then w is of infinite order, as is a_i, hence $w \mapsto a_i$ is an isomorphism and $K_w(T^*)$ results from $K(T^*)$ by a series of HNN extensions. In particular, $K(T^*) \hookrightarrow K_w(T^*)$ by Corollary 1 to Theorem 2, and hence $K_w(T^*)$ has unsolvable word problem, which means it is not free of rank p. (Suppose on the contrary that there are free generators $x_1, \ldots, x_p$ for $K_w(T^*)$, and take a fixed set of words $a_i(x_j)$, $k_i(x_j)$ which express the original generators of $K_w(T^*)$ in terms of $x_1, \ldots, x_p$. Use these words to rewrite a given word W in terms of $x_1, \ldots, x_p$, and hence decide whether $W = 1$ by free reduction as in §4.)

In short,

$$w = 1 \text{ in } K(T^*) \Leftrightarrow K_w(T^*) = \text{free group of rank } p,$$

and we have reduced the word problem for $K(T^*)$ to a special case of the isomorphism problem; the latter problem is therefore unsolvable. □

15. Remarks. (1) The reduction of the word problem to a special case of the isomorphism problem might reinforce the impression that the isomorphism problem is more difficult. But don't forget that our group $K(T^*)$ is able to simulate the universal Turing machine—any question which reduces to a question of the universal machine's halting for a certain input can be reduced in turn to a question whether a certain word equals 1 in $K(T^*)$. *Any isomorphism question is capable of such reduction.* This follows from the theorem of Tietze [**1908**] that any finite presentation of a group can be converted into any other by a finite sequence of elementary transformations, now known as Tietze transformations. Given two presentations G_1, G_2, a universal Turing machine can systematically apply all finite sequences of Tietze transformations to G_1, and halt if and when G_2 is obtained.

Thus the isomorphism problem can also be reduced to the word problem for $K(T^*)$, and we can say that they are of the same *degree of unsolvability*.

(2) Tietze transformations are also important in the Markov [**1958**] proof of the unsolvability of the homeomorphism problem. Over-simplifying slightly, we can say that Markov constructs a 4-dimensional manifold $M(G)$, with fundamental group G, from a finite presentation G. He then shows that if G' results from G by a Tietze transformation then $M(G')$ is homeomorphic to $M(G)$. Thus if G_1 and G_2 are isomorphic, $M(G_1)$ and $M(G_2)$ are homeomorphic. Of course, if G_1 and G_2 are not isomorphic then $M(G_1)$ and $M(G_2)$, having different fundamental groups, are not homeomorphic. So,

$$G_1 \text{ isomorphic to } G_2 \Leftrightarrow M(G_1) \text{ homeomorphic to } M(G_2)$$

and a solution to the homeomorphism problem would yield a solution to the isomorphism problem, whence the homeomorphism problem is unsolvable.

(3) Suppose we let T be *any* Turing machine, not necessarily universal, again let T' be the modification which imitates T except that it erases the whole tape before halting on □, let T^* be the corresponding set of pair transformations, and let $K(T^*)$ be the group constructed to simulate T^*. Then there is a similar argument leading from the special halting problem in §7 to

THEOREM 4′. *The problem of deciding, for any T and w, whether $w = 1$ in $K(T^*)$, is unsolvable.*

Theorem 5 follows equally well from Theorem 4′, hence the unsolvability of the isomorphism problem and the homeomorphism problem depends only on the tiny amount of Turing machine theory needed to construct T' from T, not on a universal machine.

The universal machine is, of course, crucial in obtaining a *specific* group $K(T^*)$ with unsolvable word problem, and this unfortunately makes it impractical to write down a presentation of $K(T^*)$, since all known universal machines have at least 20 quintuples (see e.g. Minsky [**1967**]).

(4) You may be wondering what happened to the knot problem which was the inspiration for this whole story. It has been *solved* (!) just recently; see Waldhausen [**1978**] and Hemion [**1979**].

BIBLIOGRAPHY

Adian, S. I., 1957: *The unsolvability of certain algorithmic problems in the theory of groups* Trudy Moskov. Mat. Obsc. **6**, 231–298. (Russian)

Artin, E., 1926: *Theorie der Zöpfe*, Abh. Math. Sem. Univ. Hamburg **4**, 47–72.

Boone, W. W., 1954 a, b; 1955 a, b: *Certain simple, unsolvable problems in the theory of groups*. I, II, III, IV, Nederl. Akad. Wetensch. Proc. Ser. A. **57**, 231–237, 492–497; **58**, 252–256, 571–577.

______, 1959: *The word problem*, Ann. of Math. (2) **70**, 207–265.

Britton, J. L., 1958: *The word problem for groups*, Proc. London Math. Soc. **8**, 493–506.

______, 1963: *The word problem*, Ann. of Math. (2) **77**, 16–32.

Cohen, D. E., and Aanderaa, S., 1980: *Modular machines, the word problem for finitely presented groups, and Collins' theorem*, Word Problems II (S. I. Adian, W. W. Boone, G. Higman, eds.), North-Holland, Amsterdam, pp. 1–16.

Crossley, J. N., 1975: *Reminiscences of logicians*, Algebra and Logic (J. N. Crossley, ed.), Lecture Notes in Math., vol. 450, Springer-Verlag, Berlin and New York, 1–62.

Davis, M., 1965: *The undecidable*, Raven Press, New York.

Dehn, M., 1910: *Über die Topologie des dreidimensional Raumes*, Math. Ann. **69**, 137–168.

______, 1914: *Die beiden Kleeblattschlingen*, Math. Ann. **75**, 402–413.

Gödel, K., 1931: *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme*. I, Monatsh. f. Math. u. Phys. **38**, 173–198.

Hemion, G., 1979: *On the classification of homeomorphisms of* 2*-manifolds and the classification of* 3*-manifolds*, Acta Math. **142**, 123–155.

Hermes, H., 1961: *Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit*, Springer-Verlag, Berlin and New York.

Higman, G., 1961: *Subgroups of finitely presented groups*, Proc. Roy. Soc. Ser. A. **262**, 455–475.

Higman, G., Neumann, B. H. and Neumann, H., 1949: *Embedding theorems for groups*, J. London Math. Soc. **24**, 247–254.

Kleene, S. C., 1979: *Origins of recursive function theory*, IEEE 20th Annual Symposium on Foundations of Computer Science, pp. 371–382.

Klein, F., 1882: *Letter to Poincaré*, 14 *May* 1882, Ges. Math. Abh. III, 615–616.

Magnus, W., 1930: *Über diskontinuierliche Gruppen mit einer definierenden Relation* (*Der Freiheitssatz*), J. Reine Angew. Math. **163**, 141–165.

______, 1932: *Das Identitätsproblem für Gruppen mit einer definierenden Relation*, Math. Ann. **106**, 295–307.

Magnus, W., Karrass, A. and Solitar, D., 1966: *Combinatorial group theory*, Interscience, New York.

Magnus, W. and Chandler, B., *History of combinatorial group theory* (to appear).

Markov, A. A., 1947: *On the impossibility of certain algorithms in the theory of associative systems*, Dokl. Akad. Sci. USSR **55**, 583–586. (Russian)

______, 1958: *Insolubility of the problem of homeomorphy*, Proc. Internat. Congr. Math., pp. 300–306. (Russian)

Minsky, M., 1961: *Recursive unsolvability of Post's problem of 'Tag' and other topics in the theory of Turing machines*, Ann. of Math. (2) **74**, 437–445.

______, 1967: *Computation: finite and infinite machines*, Prentice-Hall, Englewood Cliffs, N. J.

Neumann, B. H., 1954: *An essay on free products of groups with amalgamation*, Philos. Trans. Roy. Soc. London Ser. A. **246**, 503–554.

Nielsen, J., 1921: *Om Regning med ikke kommutative Faktoren og dens Anvendelse i Gruppeteorien*, Mat. Tidsskr. B, 77–94.

Novikov, P. S., 1955: *On the algorithmic unsolvability of the word problem in group theory*, Trudy Mat. Inst. Steklov **44**, 143 pp. (Russian)

______, 1958: *Über einige algorithmische Probleme der Gruppentheorie*, Jber. Deutsch. Math. Verein **61**, 88–92.

Poincaré, H., 1892: *Sur l'Analysis situs*, Comptes Rendus **115**, 633–636.

______, 1895: *Analysis situs*, J. de l'École Polytech. (2) **1**, 1–123.

Post, E., 1936: *Finite combinatory processes—formulation*. I, J. Symbolic Logic **1**, 103–105.

______, 1946: *A variant of a recursively unsolvable problem*, Bull. Amer. Math. Soc. **52**, 264–268.

______, 1947: *Recursive unsolvability of a problem of Thue*, J. Symbolic Logic **12**, 1–11.

Rabin, M. O., 1958: *Recursive unsolvability of group theoretic problems*, Ann. of Math. (2) **67**, 172–194.

Schreier, O., 1927: *Die Untergruppen der freien Gruppen*, Abh. Math. Sem. Univ. Hamburg **5**, 161–183.

Schupp, P. E., 1974: *Some reflections on HNN extensions*, Proc. 2nd Internat. Conf. Theory of Groups (M. F. Newman, ed.), Lecture Notes in Math., vol. 372, Springer-Verlag, Berlin and New York, pp. 611–632.

Stillwell, J. C., 1980: *Classical topology and combinatorial group theory*, Springer-Verlag, Berlin and New York.

Tietze, H., 1908: *Über die topologischen Invarianten mehrdimensionaler Mannigfaltigkeiten*, Monatsh. f. Math. u. Phys. **19**, 1–118.

Turing, A. M., 1936: *On computable numbers, with an application to the Entscheidungsproblem*, Proc. London Math. Soc. (2) **42**, 230–265.

______, 1950: *The word problem in semigroups with cancellation*, Ann. of Math. (2) **52**, 491–505.

van der Waerden, B. L., 1948: *Free products of groups*, Amer. J. Math. **70**, 527–528.

Waldhausen, F., 1968: *The word problem in fundamental groups of sufficiently large irreducible* 3-*manifolds*, Ann. of Math. (2) **88**, 272–280.

______, 1978: *Recent results on sufficiently large* 3-*manifolds*, Proc. Sympos. Pure Math., vol. 32, Part 2, Amer. Math. Soc., Providence, R. I., pp. 21–38.

Department of Mathematics, Monash University, Clayton, Victoria 3168 Australia