wherever it occurs. Conversely if this cannot be done then the wff is unsatisfiable, i.e., its negation is a theorem.

There are only a finite number of sets of assignments to the atomic formulas which make M true; each row of the table must be filled with one of these sets. Further, there will be many restrictions in the table, e.g. the position {row $x = a$ and $y = b$, column Gxy} must have the same truth-value as the position {row $x = b$ and $y = a$, column Gyx}. By considering these restrictions and the finite number of row assignments one may be able to prove the impossibility of filling the table, or conversely give an algorithm whereby the infinite table may be filled. In the case of certain prefixes one can give a set of rules which must result in one or the other of these things happening, i.e., we have solved the decision problem for this prefix. These kinds of techniques are what the author refers to as "proving theorems by pattern recognition."

The first of the author's papers introduces these ideas by considering specific examples and gives a pattern recognition procedure which completely solves the already known decidable A_1E case (for satisfiability, i.e., E_1A for provability). He also describes a program which the author suggests could form a preliminary to a pattern recognition program. This program reduces a problem to a series of sub-problems with simpler prefixes and solves those sub-problems which fall into the EA satisfiability class. It is an extension of a program previously described by the author (XXX 249(2)).

The detailed theoretical considerations are contained in the author's second paper. The first section has some preliminary definitions and discussion on Herbrand expansions and lists the principal known decidable classes. The author then considers the pattern recognition technique in connection with each of these decision classes. In some cases he gives a complete pattern recognition decision algorithm and in other cases merely indicates with an example how the technique could be employed. He also briefly considers a semi-decision procedure for the $A_2E_1A_1$ reduction class. The unsettled $A_1E_1A_1$ case is discussed and the author gives what may be a decision procedure, but has only been shown to terminate for certain special cases. The relation between this last case and a game of "two-dimensional dominoes" is discussed. The paper concludes with the discussion of some simple examples from specific mathematical disciplines where the techniques could be useful. DAVID C. COOPER

HAO WANG. *Mechanical mathematics and inferential analysis.* ***Computer programming and formal systems,*** edited by P. Braffort and D. Hirschberg, Studies in logic and the foundations of mathematics, North-Holland Publishing Company, Amsterdam 1963, pp. 1–20.

This paper consists for the most part of speculations on the importance of mechanizing mechanical thinking, of a review of the author's achievements, and of possible directions for further explorations. Much of the material is repeated word for word from two of the author's previous papers (XXX 249(2) and XXXII 119(2)) but is presented here in a more lucid form. The last section of the paper reproduces in part notes of lectures given at Oxford on Herbrand's original dissertation. DAVID C. COOPER

HAO WANG. *The mechanization of mathematical arguments.* ***Experimental arithmetic, high speed computing and mathematics,*** Proceedings of symposia in applied mathematics, vol. 15, American Mathematical Society, Providence 1963, pp. 31–40.

This paper contains more general comment by the author on theorem proving by machine, on mechanical proof checking, and on proof formalizing (i.e., filling in gaps in sketched out proofs). He suggests some areas in which he thinks the theoretical work has advanced far enough so that particular programs can now be written. Known semi-decision procedures for the predicate calculus are discussed and some ideas put forward for improving them. The importance of finding decision procedures in areas in which interesting mathematical theorems may be stated is discussed. DAVID C. COOPER

J. HARTMANIS and R. E. STEARNS. *On the computational complexity of algorithms.* ***Transactions of the American Mathematical Society,*** vol. 117 (1965), pp. 285–306.

This paper contains basic results in the field of a machine-oriented theory of the complexity of computational processes. The time (i.e., the number of operations) which is necessary to obtain partial results in a process is here taken as a measure of the complexity. The main problem which is investigated in detail concerns the generation of infinite binary sequences. Such a sequence is to be produced by the binary output of an autonomous, multitape Turing machine, where it is supposed that, within an operation, the output may produce either nothing or a finite binary sequence of an a priori bounded length.

A computable binary sequence α belongs to the "complexity class" S_T—where T is a "time function," i.e., a monotone computable function which increases in at least a linear way—if there is a (multitape) Turing machine which is able to produce α in such a way that the nth member of α appears within the first $T(n)$ operations of the machine.

The main results are as follows. Every class S_T is recursively enumerable, but not recursive; there is a countable infinity of distinct classes and every computable sequence α belongs to some class. The "speed-up" theorem says that the generation of every sequence can indefinitely be speeded up, since, for every $\varepsilon > 0$, $S_T = S_{[\varepsilon T]}$ ($[\varepsilon T]$ is the smallest integer $\geqq \varepsilon T$). The class corresponding to $T(n) = n$ is thus the smallest complexity class. A sufficient condition for S_{T_1} and S_{T_2} to be equal is that the limit of the ratio $T_1(n)/T_2(n)$ as $n \to \infty$ shall be finite and $\neq 0$. (The general problem of the equality of complexity classes is however recursively unsolvable.) A sufficient condition for S_{T_1} and S_{T_2} to be distinct is proved, under the supposition that T_1 and T_2 are real-time countable in the sense of Yamada (Theorem 9). Moreover, if

$$\lim_{n \to \infty} \frac{[T_1(n)]^2}{T_2(n)} = 0,$$

then S_{T_1} is a proper part of S_{T_2}. (Again, the general inclusion problem for complexity classes is recursively unsolvable.)

The authors pay special attention to the influence of various changes of the basic model of Turing machine on the content of the complexity classes. First of all, if the machine is able to produce at most one binary symbol within each operation, then the classes change only "up to ε": If $\alpha \in S_T$ with respect to the original model, then $\alpha \in S_{[(1+\varepsilon)T]}$ with respect to the new one. Further, the complexity classes remain exactly the same, if more than one head is allowed for each tape. The effect of adding more tapes cannot reduce the time necessary for the generation of a sequence by more than the square root of the original time, since if α can be generated within $T(n)$ by a multitape machine, then it can be generated within $[T(n)]^2$ by a single-tape machine (Theorem 6). An analogous result holds for the passage from machines with n-dimensional tapes to machines with one-dimensional tapes.

There are interesting results concerning the generation of real numbers, i.e., of their binary expansions. According to Theorem 11, all algebraic numbers can be generated within the time $T(n) = n^2$. Curiously enough, however, there are transcendental numbers which require the time $T(n) = n$ only.

When provided with an input, then instead of the purpose of generating infinite (binary) sequences, the basic model of multitape Turing machine can be used in a familiar way to recognize sets of words from a finite input alphabet (languages). Every time function $T(n) \geqq n$ then determines the class of languages R which are T-recognizable in the sense that there is a machine which, for every input word w of length n, produces the nth binary output symbol within $T(n)$ operations, this symbol being 1 if and only if $w \in R$. The majority of the previous results concerning classification of infinite sequences can then be carried over with minor modifications to the case of recognition of languages (Theorem 13). Interesting problems arise concerning the mutual relation between complexity classes of languages and their classification into finite-state, context-free, etc. The authors provide a concrete example of a context-free language which is not T-recognizable with $T(n) = n$, i.e., in real time.

Some unsolved problems are mentioned, among them the important question, under what conditions can a time function T yield more than a mere estimate of the complexity of a sequence $\alpha \in S_T$. The proofs in the paper are given in an informal way and require the reader's experience with the construction of complex machines. JIŘÍ BEČVÁŘ

J. HARTMANIS and R. E. STEARNS. *Computational complexity of recursive sequences.* ***Switching circuit theory and logical design, Proceedings of the Fifth Annual Symposium, Princeton University, Princeton, N.J., November 11–13, 1964,*** The Institute of Electrical and Electronics Engineers, Inc., New York 1964, pp. 82–90.