# A Computing Procedure for Quantification Theory

The hope that mathematical methods employed in the investigation of formal logic would lead to purely computational methods for obtaining mathematical theorems goes back to Leibniz and has been revived by Peano around the turn of the century and by Hilbert's school in the 1920's. Hilbert, noting that all of classical mathematics could be formalized within quantification theory, declared that the problem of finding an algorithm for determining whether or not a given formula of quantification theory is valid was the central problem of mathematical logic. And indeed, at one time it seemed as if investigations of this “decision” problem were on the verge of success. However, it was shown by Church and by Turing that such an algorithm can not exist. This result led to considerable pessimism regarding the possibility of using modern digital computers in deciding significant mathematical questions. However, recently there has been a revival of interest in the whole question. Specifically, it has been realized that while no decision procedure exists for quantification theory there are many proof procedures available—that is, uniform procedures which will ultimately locate a proof for any formula of quantification theory which is valid but which will usually involve seeking “forever” in the case of a formula which is not valid—and that some of these proof procedures could well turn out to be feasible for use with modern computing machinery. Hao Wang [9] and P. C. Gilmore [3] have each produced working programs which employ proof procedures in quantification theory. Gilmore's program employs a form of a basic theorem of mathematical logic due to Herbrand, and Wang's makes use of a formulation of quantification theory related to those studied by Gentzen. However, both programs encounter decisive difficulties with any but the simplest formulas of quantification theory, in connection with methods of doing propositional calculus. Wang's program, because of its use of Gentzen-like methods, involves exponentiation on the total number of truth-functional connectives, whereas Gilmore's program, using normal forms, involves exponentiation on the number of clauses present. Both methods are superior in many cases to truth table methods which involve exponentiation on the total number of variables present, and represent important initial contributions, but both run into difficulty with some fairly simple examples. In the present paper, a uniform proof procedure for quantification theory is given which is feasible for use with some rather complicated formulas and which does not ordinarily lead to exponentiation. The superiority of the present procedure over those previously available is indicated in part by the fact that a formula on which Gilmore's routine for the IBM 704 causes the machine to computer for 21 minutes without obtaining a result was worked successfully by hand computation using the present method in 30 minutes. It should be mentioned that, before it can be hoped to employ proof procedures for quantification theory in obtaining proofs of theorems belonging to “genuine” mathematics, finite axiomatizations, which are “short,” must be obtained for various branches of mathematics.