Final Report

ON

MATHEMATICAL PROCEDURES FOR DECISION PROBLEMS

Under Technical Supervision of

Commanding General, Aberdeen Proving Ground

Work performed during period 1 June 1954 to 31 August 1954

Under Contract No. DA-36-034-ORD-1645

Department of Army Project No. 599-01-004

Ordnance Research and Development Project No. TB2-0001

Office of Ordnance Research Project No. 1333

Martin Davis

Institute for Advanced Study

Princeton, New Jersey

October 1954

ON

MATHEMATICAL PROCEDURES FOR DECISION PROBLEMS

A Program for Presburger's Algorithm for Additive Number Theory on the Institute for Advanced Study Digital Computer

Part I

GENERAL CONSIDERATIONS

1. Introduction. Much work has recently been done on the existence or non-existence of decision procedures for various logistic systems.[1] It has been found that any logistic system which is, in a suitable precise sense, "adequate" for elementary number theory has an unsolvable decision problem.[2] However, Presburger has shown that if we restrict ourselves to that fragment of elementary number theory which involves addition only (not multiplication), a decision procedure does exist.[3] Decision procedures for other logistic systems are also known. Of particular interest is Tarski's decision procedure for elementary algebra.[4]

This paper reports on the author's program, written for the Institute for Advanced Study electronic digital computer (hereafter abbreviated IASC), for Presburger's decision procedure. This program enables the IASC to furnish the truth value of any proposition (i.e., "truth", if the proposition is true, "falsehood", if it is false) from elementary additive number theory,[5] (subject, of course, to the usual limitations of time and memory space). We shall also touch briefly on the more general question of the programming of problems of a purely logical nature for digital computers.

The author would like to express his appreciation for invaluable assistance to Dr. Herman Goldstine and Mr. James Cooley of the IAS computer group.

2. Notation. We assume that the reader is familiar with the basic operations of elementary logic. In particular, we use the notation "$\sim$" for negation, "&"

for conjunction, "$\vee$" for (inclusive) alternation, and "(Ex_i)", where x_i is an individual variable for existential quantification. Following the practice of the Polish school we write binary connectives preceding the symbols on which they operate. Thus $\& AB$ renders "A and B." As is well known, the other operations of elementary logic are representable in terms of these. Thus, "A implies B" is rendered by $\vee \sim AB$, and universal quantification by $\sim (Ex_i) \sim$.

The variables $x_1, x_2, x_3, \ldots,$ are to be thought of as ranging over the integers positive, negative, or zero. A proposition belongs to <u>additive elementary number theory</u> if it can be written in terms of 0, 1, +, =, the variables x_i, and the operations of elementary logic. E.g.,

$$\sim (Ex_1) \sim \; \sim (Ex_2) \sim \vee \sim \& (Ex_3)\, x_1 = x_3 + x_3\, (Ex_3)\, x_2 = x_3 + x_3\, (Ex_3)\, x_1 + x_2 = x_3 + x_3,$$

represents the proposition:

The sum of any two even numbers is even.

Hence, this proposition belongs to additive number theory. As usual, if α, β are any expressions, we write $\alpha \neq \beta$ for $\sim(\alpha = \beta)$. We note that congruence modulo any definite integer n is representable in additive number theory. For,

$$\alpha \equiv \beta \pmod n, \text{ if and only } (Ex_i)\,(\alpha = \beta + \underbrace{x_i + x_i = \ldots + x_i}_{n \text{ terms}}).$$

(Here, if α and β are thought of as symbolic expressions, rather than definite integers, i must be chosen so that x_i does not occur in either α or β.)

Presburger's algorithm is an inductive one which, beginning with a proposition written in terms of the original notions of additive elementary number theory, introduces congruences and negations of congruences (or as we shall say, <u>incongruences</u>). Hence, we permit $\equiv n$ and $\not\equiv n$ among our notions. Also, if $n > 0$ is a definite integer, we write nx_i for the expression $\underbrace{x_i + x_i + \ldots + x_i}_{n \text{ terms}}$ and n for the expression $\underbrace{1 + 1 + \ldots + 1}_{n \text{ terms}}$. We call an expression of the form: