

DOING LOGIC BY COMPUTER

RICHARD L. PURTILL

Teachers of logic often tell their classes that a great number of tasks in logic can be performed mechanically. Surprisingly enough, most logicians have failed to make full use of existing mechanical devices which could demonstrate this point, and by so doing, increase the interest of students in the theory and application of logic.

In this paper¹, I will first describe some simple ways of using digital computers to do certain tasks connected with logic, for example, draw up truth tables, decide whether statements are tautologies, contingent or contradictory and whether arguments are valid or invalid. I will then go on to discuss some extensions of these techniques. The advantage of the techniques I am about to describe is that they can be used on most machines which can be programmed by means of FORTRAN, which is a "programming language" in which instructions can be written in combinations of English and algebraic statements. Many such machines are in use including some small computers used mainly for bookkeeping operations. There are a few specially designed logical computers in existence, and some rather rare and expensive computers have certain logical capabilities in addition to their mathematical ones. But such computers exist in fairly small numbers, and are fairly difficult of access, whereas machines which can be programmed in FORTRAN are fairly common, and more likely to be available.

Let us begin with a very simple problem in logic. A and B are true statements; X and Y are false statements. Our problem is to discover the truth or falsity of certain compound statements made up of A 's, B 's, X 's and Y 's joined by the truth functional connectives "." ("and") "v," ("or"). Now the average computer has no such special symbols as, "." and "v" and it is designed to do mathematical rather than logical calculations. What

1. Part of this research was carried out with the aid of several grants made by the Research Office of Western Washington State College. I wish to thank the following former students who contributed ideas and help: David Ault, Larry Lingbloom and Richard Thompson.