Notre Dame Journal of Formal Logic
Volume XX, Number 2, April 1979
NDJFAM

THE DECIDABILITY OF ONE-VARIABLE PROPOSITIONAL CALCULI

M. D. GLADSTONE

"Propositional calculus" (or "**PC**") will be defined more precisely later on. For the moment it is enough to say that the meaning is the usual one, with the qualifications

(i) the set of axioms is finite (but need not be tautologous),
(ii) the rules of inference are substitution and modus ponens, i.e., A, $A \supset B \vdash B$, where "$\supset$" *may* stand for a combination of two or more logical connectives.

Let a **PC** be *monadic* (*diadic*) iff every axiom contains at most one (two) distinct variable(s). A general discussion of such systems will be found in [1]. In [3], Hughes constructs a diadic **PC** with a non-recursive class of theorems. As we shall see, this cannot be done for monadic **PC**s. In fact the object of the present paper is to describe a single algorithm for testing theoremhood in any given monadic **PC**. Some similarity will appear between monadic **PC**s and a certain type of combinatorial system studied by Post in [5].

We begin by investigating an offshoot of Post's system to be called an **L**-*system* (**L** for "left").

Definition: **L**-*System*. An **L**-system π consists of

(i) a countable non-empty alphabet $\mathfrak{A}_\pi$;
(ii) a finite set of ordered pairs, known as *rules*, of the form (Γ,B), where Γ is a finite set of words on $\mathfrak{A}_\pi$ and B is a word on $\mathfrak{A}_\pi$; the members of Γ are the *premises* of the rule, and B is the *conclusion.*

$\wedge$ is the empty word. $\emptyset$ is the empty set.

I assume the reader is familiar with the general notion of "proof tree" (if not, see [4] for instance). In the present case we describe a finite set of words of π in tree array as a "proof tree in π" iff, for every word Z having n (≥ 1) words immediately above it, there exist a word X and rule

Received March 29, 1977