Notre Dame Journal of Formal Logic
Volume XIX, Number 2, April 1978
NDJFAM

A NOTE ON TURING MACHINE REGULARITY AND PRIMITIVE RECURSION

NICHOLAS J. de LILLO

1 *Introduction* The purpose of this paper is to present an explicit Turing machine **Z** which computes any function which is defined by means of primitive recursion from two given computable functions. The formulation of **Z** uses results of Davis [1] and Mal'cev [3], with the added feature that **Z** yields outputs in a standard form, such outputs usable as inputs in subsequent Turing machines which can be activated after **Z** has completed its computation. Such machines as **Z** are defined as n-regular, for a positive integer n. The course of a computation in **Z** follows along lines suggested by Davis [2], for a similar computation using abstract programs instead of Turing machines.

2 *Preliminary concepts* We will assume a general familiarity with [1], explicitly defining only those concepts which are absolutely necessary for the continuity of this discussion. A Turing machine[1] is any non-empty and finite set of quadruples, any one of which assumes the form (i) $q_i \mathsf{S}_j \mathsf{S}_k q_l$, or (ii) $q_i \mathsf{S}_j \mathsf{R} q_l$, or (iii) $q_i \mathsf{S}_j \mathsf{L} q_l$, where i, j, k, l are positive integers. The symbols q_i, q_l are elements of a finite set Q, called the *internal states* of the machine; the symbols S_j, S_k are elements of the set $A = \{1, B\}$ disjoint from Q and called the alphabet of the machine; the symbols L and R are distinct symbols not in $Q \cup A$. It is understood that no two distinct quadruples of a given Turing machine begin with the same first two symbols. The usual meanings are attached to the quadruples: (i) is the instruction which, when the state of the machine is q_i and the symbol S_j is being scanned, erases S_j and prints S_k in its place, the machine then moving to state q_l; (ii) instructs the machine to move one square to the right and change to state q_l when the machine is in state q_i and scans a square with S_j printed there; (iii) is the instruction similar to (ii), except the machine moves one square to the left.

1. Using the terminology of [1], this paper will deal only with *simple* Turing machines, but these results can easily be generalized to the case of relative computability.

Received December 18, 1974