Notre Dame Journal of Formal Logic
Volume X, Number 3, July 1969

NOTE ON THE MORTALITY PROBLEM FOR SHIFT STATE TREES

MICHAEL ANDERSON

In [1] the problem of determining if a Turing machine is mortal from its state transition structure was considered. A state tree was defined to be mortal if it corresponded only to mortal Turing machines. Necessary and sufficient conditions were derived for the mortality of any state tree. We give here a result on the corresponding problem for shift state trees, state trees which retain the shift structure of the machine.

The following is a generalization of Lemma 1 of [1]:

Lemma: *For any shift state tree T, path $(q_1, \ldots, q_i)$ in T, and input I; there is a Turing machine* M *with shift state tree T and an input I' such that when* M *is applied to I' it reaches state q_i on I. The only print instructions determined by these conditions on* M *are the (q_j, x_j) instructions where $(x_1, \ldots, x_{i-1})$ is the input sequence of the given path.*

The result follows trivially by induction on the length of the path.

Define a state tree to be shift mortal if there is a mortal shift state tree which corresponds to it, i.e. if shift instructions can be added to the state tree to make it a mortal shift state tree.

Theorem: *A state tree is shift mortal iff every terminal point of the state tree is a q_0 or a q_i whose cycle is not a lower path.*

Proof: Let T be a state tree which has a terminal point q_i whose cycle is a lower path. Let $(q_1, \ldots, q_i)$ be the path in T leading to the initial point of the cycle. The only state which occurs in both this path and the cycle path is q_i. Let $\mathfrak{T}$ be a shift state tree corresponding to T. Apply the lemma to $\mathfrak{T}$, $(q_1, \ldots, q_i)$, and the blank tape. Choose the undetermined print instructions of the resulting machine to be blanks. Then M has an endless computation on I'.

For the converse we can prove the following stronger result: *A Turing machine which shifts only to the right (left) is mortal if every terminal point of its state tree is either a q_0 or a q_i whose cycle is not a lower path.*

Received December 2, 1968

Let **R** be a right shifting Turing machine which does not halt on some input I. Since I has only a finite number of strokes and a right shifting machine never scans any cell twice, **M** will eventually scan only blank cells. Thus eventually its state transitions will take place only on lower edges of its state tree. Thus the state tree of **R** will have a terminal point whose cycle is a lower path.

The result of the theorem is to reduce the mortality problem for shift state trees to the problem of determining which shift structures on shift state trees give mortal shift state trees. This part of the problem seems to be non-trivial.

We note that the shift mortal state trees can be formally represented by adding to the system **K** of [1] the following rule

Designation: *For any occurrence of* q_0 *in a thesis* θ *other than as the last symbol in* θ*, we may substitute a* q_1

This rule makes A_2 superfluous.

REFERENCES

[1] Anderson, M., "Approximation to a decision procedure for the halting problem," *Notre Dame Journal of Formal Logic*, vol. IX (1968), pp. 305-312.

Seminar in Symbolic Logic
University of Notre Dame
Notre Dame, Indiana