The Solvability of the Halting Problem for 2-State Post Machines

STÅL AANDERAA

Oslo University, Oslo, Norway

AND

PATRICK C. FISCHER

Cornell University, Ithaca, New York

ABSTRACT. A Post machine is a Turing machine which cannot both write and move on the same machine step. It is shown that the halting problem for the class of 2-state Post machines is solvable. Thus, there can be no universal 2-state Post machine. This is in contrast with the result of Shannon that there exist universal 2-state Turing machines when the machines are capable of both writing and moving on the same step.

This paper is directed to one of the principal differences between the definition of Turing machines in terms of quintuples as given by Turing in [5], and the quadruple version given by Post in [3] and used by Davis [1]. The halting problem for the class of all 2-state (quintuple version) Turing machines is unsolvable, since Shannon has exhibited a universal 2-state Turing machine [4]. We show here that the halting problem for the class of all 2-state Post machines is solvable, i.e., that there exists a uniform effective procedure for determining, given any 2-state Post machine and its initial instantaneous description, whether or not the ensuing computation will halt. This result is mentioned, but not proved, in [2].

We assume familiarity with at least one of the references given above, and summarize the Post formalism only briefly. A Post machine is a variant of a Turing machine involving the usual finite set Σ of symbols, finite set Q of states, and unbounded almost-blank tape with read-write head. It is described as a set of quadruples of the form $\langle q_i, S_j, X, q_k\rangle$ where $q_i \in Q$, $q_k \in Q$, $S_j \in \Sigma$, and either $X \in \Sigma$ or $X = L$ or $X = R$ $(L, R \notin \Sigma)$. No two distinct quadruples may agree in their leftmost two components. Let us denote quadruples in which $X \in \Sigma$ as *S-quadruples* and those in which $X = L$ or R as *M-quadruples*.

The membership of the quadruple $\langle q_i, S_j, X, q_k\rangle$ in the set defining a Post machine means that during a computation if the machine is in state q_i while scanning the symbol S_j, it takes action X and enters state q_k. If action X is a symbol, then the machine replaces the symbol S_j under scan by the symbol denoted by X; if $X = L$ or R, the machine moves one square to the left or right, respectively, on its tape. We call such an action the *execution* of the quadruple. The machine halts if it encounters a state-symbol configuration for which it has no quadruple beginning with that configuration.

The solvability of the halting problem requires exploitation of the 2-state assumption and of the fact that a Post machine, unlike a quintuple Turing machine, can-

Journal of the Association for Computing Machinery, Vol. 14, No. 4, October 1967, pp. 677–682

not both write a symbol and move during the execution of a single quadruple. (The halting problem for 3-state Post machines, however, is unsolvable [2].) We begin the analysis with some lemmas.

Definition. A *blocking loop* is a (necessarily finite) sequence of S-quadruples of the form:

$$\langle q_{i_1}, S_{j_1}, S_{j_2}, q_{i_2}\rangle$$
$$\langle q_{i_2}, S_{j_2}, S_{j_3}, q_{i_3}\rangle$$
$$\cdots$$
$$\langle q_{i_n}, S_{j_n}, S_{j_1}, q_{i_1}\rangle$$

The length of a blocking loop is the number n of quadruples in the loop.

LEMMA 1. *The halting or nonhalting of a Post machine is unaltered if the quadruples in a blocking loop of length $n > 1$ are replaced by blocking loops of length 1 as follows:*

$$\langle q_{i_1}, S_{j_1}, S_{j_1}, q_{i_1}\rangle$$
$$\langle q_{i_2}, S_{j_2}, S_{j_2}, q_{i_2}\rangle$$
$$\cdots$$
$$\langle q_{i_n}, S_{j_n}, S_{j_n}, q_{i_n}\rangle$$

PROOF. Once a machine enters any blocking loop, it never halts.

LEMMA 2. *For every Post machine there is an equivalent Post machine (except for the nature of any blocking loops) with the same numbers of symbols and states in which the execution of an S-quadruple during a computation is either followed by the execution of an M-quadruple or by the execution of the same S-quadruple (in which case the machine is in a length 1 blocking loop) or by a halt.*

PROOF. First apply Lemma 1 to eliminate all blocking loops of length $n > 1$. Then search for sets containing two S-quadruples and one M-quadruple of the following form, where $X = L$ or R:

$$\langle q_{i_1}, S_{j_1}, S_{j_2}, q_{i_2}\rangle$$
$$\langle q_{i_2}, S_{j_2}, S_{j_3}, q_{i_3}\rangle$$
$$\langle q_{i_3}, S_{j_3}, X, q_k\rangle$$

Change the first quadruple in such a set to $\langle q_{i_1}, S_{j_1}, S_{j_3}, q_{i_3}\rangle$. The ultimate behavior of the machine is not affected by this change, and the execution of the first quadruple will now be followed by the execution of an M-quadruple. This process is repeated until all S-quadruples not followed by themselves are followed by M-quadruples.

We now restrict our attention to machines satisfying the conclusion of Lemma 2. We call such machines *active* Post machines.

Definition. If a Post machine contains an S-quadruple $\langle q_i, S_j, S_t, q_k\rangle$, then S_t will be said to be a *successor* to S_j. A symbol which is a successor to some symbol will be called a *successor symbol.*

LEMMA 3. *In a 2-state active Post machine, any symbol which is a successor symbol has at most one successor.*

PROOF. Let S_t be a successor symbol, and $\langle q_i, S_j, S_t, q_k\rangle$ an S-quadruple producing S_t. By Lemma 2 a quadruple beginning with $\langle q_k, S_t \rangle$ must be an M-quadruple. Since there are at most two quadruples with S_t in the second position, there is at most one S-quadruple with S_t in that position. Thus, S_t has at most one successor.

Let us consider further a successor symbol such as the S_t in Lemma 3. If the symbol is produced during a computation on a given square of the tape of an active Post machine P, the next quadruple executed must be an M-quadruple. Let us assume the M-quadruple in question is $\langle q_k, S_t, R, q_p \rangle$. Now suppose the read-write head of P eventually returns from the right to the square containing S_t. If P is in state q_k at this time, it will merely "bounce off" the square, leaving the S_t unchanged. The only way S_t can be changed or P can cross the square containing S_t is for P to enter the square in state $\bar{q}_k$, the state which is not q_k. If this happens, we say that P *penetrates* the square.

It can now be seen that 2-state Post machines are "locally deterministic" in the following sense:

LEMMA 4. *The contents of a given square on the tape of an active 2-state Post machine depend only upon the symbol initially in the square, the state of the machine the first time the square is visited, and the number of times the square is subsequently penetrated.*

PROOF. After the first visit to a given tape square, the symbol written on it cannot be changed except by penetrating the square, and a given penetration can occur in only one way.

Let us now assume that the tape squares of a 2-state active Post machine P are indexed by the integers as though they were unit intervals along the x-axis. Let l and r be chosen so that the tape segment T_M between and including squares l and r contains all the squares initially nonblank and the square being scanned. Thus squares $l-1$, $l-2$, $l-3$, $\cdots$ form a one-ended, unbounded, initially-blank tape segment T_L to the left, and squares $r+1$, $r+2$, $r+3$, $\cdots$ form such a segment T_R to the right. We analyze the potential behavior of P during penetrations of T_R by observing its behavior on squares $r+1$ through $r+4$; any properties developed will, of course, apply in symmetric fashion to T_L.

LEMMA 5. *If P eventually visits square $r+4$, its state when it first visits square $r+4$ will be the same as its state when it first visits square $r+2$.*

PROOF. Squares $r+1$ through $r+4$ are all initially blank. Assume that P is in state q when it first visits square $r+1$.

Case 1. *P is in state q when it first visits square $r+2$.* Then we know that when P first visits the leftmost square of an unbounded blank tape segment while in state q, it first visits the square second from the left in state q. Now if we consider the segment $r+2$, $r+3$, $r+4$, $\cdots$ we find that its leftmost square is first visited while P is in state q. Thus, square $r+3$ must first be visited in state q and, similarly, square $r+4$ must first be visited with P in state q.

Case 2*a*. *P is in state $\bar{q}$ when square $r+2$ is first visited and in state $\bar{q}$ when square $r+3$ is.* Then we apply the argument of Case 1 to the segment $r+2$, $r+3$, $r+4$, $\cdots$ and conclude that P is in state $\bar{q}$ when it first visits square $r+4$.

Case 2*b*. *P is in state $\bar{q}$ when square $r+2$ is first visited and in state q when square $r+3$ is.* Then we know that if P first visits the leftmost square of an unbounded blank tape segment in state q, it first visits the square second from the left in state

$\bar{q}$. We now consider the initially blank segment $r+3$, $r+4$, $r+5$, $\cdots$. Since P visits its leftmost square initially while in state q, P is in state $\bar{q}$ when it first visits square $r+4$.

We now describe a Turing machine P' which is shown to halt if and only if P does. Squares $l-4$ through $r+4$ of the tape of P' contain the same symbols as the corresponding squares on the tape of P. Other squares of the tape of P' are used to keep track of four integers, R_1, R_2, L_1, and L_2. All four counts are initially zero. When P is in segment T_M, P' simply imitates the behavior of P. When P is in T_R, the contents of squares $r+1$ through $r+4$ are noted by P' and the following additional procedures are performed:

(1R) If P penetrates square $r+2$ from the left, a flag f_R is set to 1.

(2R) If P leaves square $r+2$ heading to the left and f_R has been set to 1, counter R_1 is increased by 1 and f_R is set to zero. (This happens whether or not square $r+1$ is actually penetrated.)

(3R) If P penetrates square $r+4$ from the left and counters R_1 and R_2 both contain the same number, then P' deliberately diverges (runs forever) by entering a blocking loop. Otherwise, a flag q_R is set to 1.

(4R) If P tries to leave square $r+4$ heading to the right, P' acts as though P were placed back on square $r+4$ in the proper state to penetrate that square.

(5R) If P leaves square $r+4$ heading to the left and g_R has been set to 1, counter R_2 is increased by 1 and g_R is set to zero. (This happens whether or not square $r+3$ is actually penetrated.)

The counter R_1 will keep count of the number of excursions into the segment $r+2$, $r+3$, $r+4$, $\cdots$. In order to avoid counting instances in which the read-write head visits square $r+2$ from the left without penetrating it, we have introduced the flag f_R. In like manner the counter R_2 will keep count of the number of excursions into the segment $r+4$, $r+5$, $r+6$, $\cdots$, and the flag g_R will be used to exclude nonpenetrating visits to square $r+4$ from the left.

A similar set of rules applies when P is in T_L. Rules (1L) through (5L) are obtained from those above in the obvious way, by interchanging "left" and "right" and by replacing f_R, g_R, R_1, R_2, $r+1$, $r+2$, $r+3$, $r+4$ by f_L, g_L, L_1, L_2, $l-1$, $l-2$, $l-3$, $l-4$, respectively.

LEMMA 6. *The machine P' halts if and only if P does.*

PROOF. The behavior of P and P' on T_M is identical. We compare the behavior of P and P' when P is in T_R.

Observe that for all i, if counter R_1 contains i, then for at least the first i excursions into the initially blank segment $r+2$, $r+3$, $r+4$, $\cdots$ via penetrating square $r+2$ from the left, P will return to square $r+1$. For $i=1$ this is true because during the first excursion into the segment counters R_1 and R_2 are both zero. Thus, if square $r+4$ is ever penetrated, rule (3R) will cause P' to diverge. If R_1 reaches a count of 1, we know that P' eventually reaches square $r+1$ without rule (3R) ever applying, so that square $r+4$ is not penetrated and rule (4R) is not used during the excursion. Consequently, the behaviors of P' and P are the same during this excursion, and P must also return to square $r+1$.

Now assume the result holds for $1 \leq i \leq k$. On the $(k+1)$-st excursion into the segment $r+2$, $r+3$, $r+4$, $\cdots$ counter R_1 contains the number k. If R_1 reaches a count of $k+1$, we know that P' eventually reaches square $r+1$ without rule $(3R)$ applying during the excursion. Thus, during any penetrations of square $r+4$ counter R_2 must contain a number less than k. By rule $(5R)$ the count in R_2 can be seen to register the number of excursions into the segment $r+4, r+5, r+6, \cdots$ made by P. By the induction hypothesis, for at least the first k excursions into an initially blank segment, we are guaranteed that P will return from the segment. Since by Lemma 5, P is in the same state when it first penetrates square $r+4$ as when it first penetrates square $r+2$, one may apply the above argument to the segment $r+4, r+5, r+6, \cdots$. We therefore find that the returns forced upon P' by rule $(4R)$ and the returns which P makes under its own power have the same effects upon the machine states and the symbols in squares $r+1$ through $r+4$. Thus, when P' returns to square $r+1$, P must do likewise.

We now see that rule $(3R)$ accurately predicts whether or not P will return from an excursion into the segment $r+4, r+5, r+6, \cdots$ whenever it penetrates square $r+4$. If the contents of R_1 are greater than those of R_2, P will return to square $r+3$ and P' will leave the proper symbols on squares $r+1$ through $r+4$. If, on the other hand, R_1 and R_2 both contain the same number i, say, then P will diverge. For this means that during the ith excursion into the segment $r+2, r+3, r+4, \cdots$, P makes its ith excursion into the segment $r+4, r+5, r+6, \cdots$. In similar fashion one can see that P makes its ith excursion into the segment $r+6, r+7, r+8, \cdots$ and in fact that for any positive integer k, it reaches and penetrates square $r+2k+2$ before it ever returns to square $r+2k$. Thus, P diverges by proceeding indefinitely along the tape to the right with only minor hesitations along the way.

Since a symmetric argument will apply when P is in T_L, we now may conclude the correctness of Lemma 6. When P is between squares $l-4$ and $r+4$, P and P' have the same behavior and one will halt if and only if the other does. If P penetrates square $r+4$ from the left, then either rule $(3R)$ applies, in which case both P and P' diverge, or it does not, in which case P and P' both end up between squares $l-4$ and $r+4$ still computing and in the same configurations. The same happens if P penetrates square $l-4$ from the right.

Since P' is constructed in a uniform effective manner from P, it now suffices to solve the halting problem for P'.

LEMMA 7. *The halting problem for machines of the form of P' is solvable.*

PROOF. Except for the infinitely many possible states of the four counters, P' has only finitely many possible (tape-state) configurations. A computation by P' must eventually either halt, diverge by entering a length 1 blocking loop, or repeat a configuration. The latter case means that there are positive integers t and k such that the basic control unit of P' is in the same state and scanning the same square, and squares $l-4$ through $r+4$ contain the same symbols at both times t and $t+k$. If a repeated configuration occurs, P' will never halt, for between times $t+k$ and $t+2k$ exactly the same actions will again occur unless the contents of the counters change in such a way as to cause rule $(3R)$ or $(3L)$ to apply, in which case the machine diverges anyhow.

It has thus been shown:

THEOREM. *The halting problem for the class of 2-state Post machines is solvable.*

REFERENCES

1. Davis, M. *Computability and Unsolvability.* McGraw-Hill, New York, 1958.
2. Fischer, P. C. On formalisms for Turing machines. *J. ACM 12*, 4 (Oct. 1965), 570–580.
3. Post, E. L. Recursive unsolvability of a problem of Thue. *J. Symbolic Logic 12* (1947), 1–11.
4. Shannon, C. E. A universal Turing machine with two internal states. In *Automata Studies*, Shannon, C. E., and McCarthy, J. (Eds.), Princeton U. Press, Princeton, N. J., 1956.
5. Turing, A. M. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.* {2}, *42* (1936–37), 230–265; Correction, *ibid.*, *43* (1937), 544–546.

RECEIVED JANUARY, 1967