A CONSTRUCTION FOR THE INVERSE OF A TURING MACHINE

BY

JAMES GIPS

STAN-CS-73-390
AUGUST 1973

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

A Construction for the Inverse of a Turing Machine

by

James Gips *
Computer Science Department
Stanford University
Stanford, California 94305

15 August 1973

*The author is supported by an I.B.M. fellowship

Abstract

A direct construct ion for the inverse of a Turing machine is presented.

PROBLEM

Given Turing machine M that for input- tape I produces output tape 0, i.e. M(I) = 0, construct an inverse Turing machine M' such that M'(0) = I. M' should be non-deterministic so that the set of output tapes of M' given 0 is exactly the set of all possible input tapes to M that would produce 0.

McCarthy [1] has investigated enumerative techniques for the inversion of the partial function defined by a Turing machine. In this paper, a simple met hod for the direct construction of M' given M is presented.

CONVENTIONS

The Turing machine conventions used are basically those of Minsky [2].

A Turing machine is a set of quintuples of the form (old-state, symbol-scanned, new-state, symbol-written, direction) or (q_g, s_h, q_j, s_k, d). The States are $q_1, q_2, \ldots q_{n-1}$, halt, with q_b the initial state. The tape symbols are $s_1, s_2, \ldots s_m$. The directions are left, right, and "-". Each quintuple is of one of three types:

(1) $(q_g, s_h, q_j, s_k,$ left) interpreted as "if in state q_g and scan symbol s_h, then enter state q_j, write symbol s_k, and move tape head left",

(2) $(q_g, s_h, q_j, s_k,$ right) interpreted as "if in state q_g and scan symbol s_h, then enter state q_j, write symbol s_k, and move tape head right".

(3) (q_g, s_h, halt, s_k, -) interpreted as "if in state q_g and scan symbol s_h, then write symbol s_k and halt in place".

An output tape of a Turing machine is a tape that exists after a quintuple of type 3 is applied.

CONSTRUCTION ALGORITHM

Given Turing machine M with m tape symbols, n states, and p quintuples, a new Turing machine M' with m tape symbols, 2n states, and 2p+c quintuples, where c is the number of quintuples of M with the initial state as the first element, is constructed. The tape symbols of M' are the same as the tape symbols of M. If the states of M are $q_1, q_2, \ldots q_{n-1}$, halt, the states of M' are begin, q_1', q_1'', q_2', q_2'', ... q_{n-1}', q_{n-1}'', halt. The initial state of M' is begin.

For each quintuple in M two quintuples are added to M' as follows:

(1) For each quintuple (q_g, s_h, q_j, s_k, left) of type 1 in M the quintuples (q_j', s_k, q_g', s_h, right) and (q_j', s_k, q,", s_h, left) are added to M'.

(2) For each quintuple (q_g, s_h, q_j, s_k, right) of type 2 in M the quintuples (q_j'', s_k, q_g', s_h, right) and (q_j'', s_k, q_g'', s_h, left) are added to M'.

(3) For each quintuple (q_g, s_h, halt, s_k, -) of type 3 in M, the quintuples (begin, s_k, q_g', s_h, right) and (begin, s_k, q,", s_h, left) are added to M'.

Additionally, one quintuple is added to M' for each quintuple in M that contains the initial state, q_b, as its first element (old-state) as follows:

(1) For each quintuple (q_b, s_h, q_j, s_k, left) Of type 1 in M, where q_b is the initial state, the quintuple (q_j', s_k, halt, s_h, -) is added to M'.

(2) For each quintuple (q_b, s_h, q_j, s_k, right) of type 2 in M, where q_b is the initial state, the quintuple (q_j", s_k, halt, s_h, -) is added to M'.

(3) For each quintuple (q_b, s_h, halt, s_k, -) of type 3 in M, where q_b is the initial state, the quintuple (begin, s_k, halt, s_h, -) is added to M'.

DESCRIPTION

The quintuples of M' are constructed by transposing the old-state and symbol-scanned of each quintuple of M with the new-state and symbol-written of that quintuple so that M' reverses the possible computations of M.

In the first part of the construction, the new-state and symbol-written of each quintuple of M becomes the old-state and symbol-scanned of two new quintuples of M'. If the new-state of the quintuple of M is halt, the old-state of the quintuples of M' is begin. If the direction of the quintuple of M is left, the old-state of the quintuples of M' is primed; if the direction is right, the old-state is double primed. The old-state and symbol-scanned of each quintuple of M becomes the new-state and symbol-written of the two added quintuples of M'. In one of the new quintuples of M' the direction is right and the new-state is primed; in the other the direction is left and the new-state is double primed. Informally, M' continually looks to the left and to the right to determine what quintuples of M could have been applied to result in the current state and tape configuration. The primes of the state of M' keeps track of which direction M' is currently looking.

In the second part of the construction, a quintuple with halt as the new-state is

added to M' for each quintuple of M with the initial state as the old-state. Again, the new-state and symbol-writ ten of the quintuple of M becomes the old-state and symbol-scanned of the quintuple of M'. If the new-state of the quintuple of M is halt, the old-state of the quintuple of M' is begin. If the direction of the quintuple of M is left, the old-state of the quintuple of M' is primed; if the direction is right, the old-state is double primed. The symbol-scanned of the quintuple of M becomes the symbol-written of the quintuple of M'. Informally, an output tape of M' is produced at each point that M could have begun a computation.

M' is designed to trace backwards through all possible computations of M that could result in output tape 0. Each computation of M' has the same length as the corresponding computation of M. Further, the sequence of tape configurations for each computation of M' is identical to the reverse of the sequence of tape configurations for the corresponding computation of M, except for the locations of the tape heads. The sequence of states for each computation of M' is equivalent to the reverse of the sequence of states for the corresponding computation of M except for the first state of each sequence (which is always the initial state) and the final state of each sequence (which is always halt) if states q_x' and q_x'' in M' are considered equivalent to state q_x in M for $1 \leq x \leq n-1$.

REMARKS

In general, the construction method does not result in quintuples in M' for every possible circumstance (i.e. every possible old-state : symbol-scanned pair). Some of

the computations of M' may run into a dead-end, a situation where no quintuple is applicable. If an inverse machine M' does run into a dead-end, it simply means that M' is not retracing a computation that could have been done by machine M. If dead-ends are displeasing then quintuples can be added to M' for every state : symbol pair not occurring as an old-state : symbol-scanned pair in M'. The new-state of these added quintuples would be an extra state that if entered results in M' looping forever. For practical use- of the construct ion, dead-ends seem desirable as their use would increase the efficiency of a serial, deterministic encoding of M'.

Some of the computations of M' may not terminate. Recall that the goal for the construction is that the set of all output tapes of M' (i.e. all tapes that exist after M' enters the halt state) for input tape 0 is exactly the set of possible input tapes for M that would result in 0 as an output tape.

EXAMPLE

As a simple example, consider Turing machine M_1 with four tape symbols: X, E, 0, and b, where b is the symbol for "blank"; three states: q_1, q_2, and halt, where q_1 is the initial state; and four quintuples:

$(q_1, X, q_2, b, right)$

$(q_2, X, q_1, b, right)$

$(q_1, b, halt, E, -)$

$(q_2, b, halt, 0, -)$

M_1 is a Turing machine that calculates the parity of the string of X's extending to the right of the initial position of the tape head, erases the X's, and writes E if the parity is even and 0 if the parity is odd.

Sample input tape for M_1:

... b b X X X X X b b ...
↑

Resulting output tape:

... b b b b b b b 0 b . . .
↑

The inverse Turing machine, M_1', has the same four tape symbols; six states: begin, q_1', q_1'', q_2', q_2'', and halt; and ten quintuples:

(q_2'', b, q_1', X, right)	from the first quintuple of M_1
(q_2'', b, q_1'', X, left)	from the first quintuple of M_1
(q_1'', b, q_2', x, right)	from the second quintuple of M_1
(q_1'', b, q_2'', X, left)	from the second quintuple of M_1
(begin, E, q_1', b, right)	from the third quintuple of M_1
(begin, E, q_1'', b, left)	from the third quintuple of M_1
(begin, 0, q_2', b, right)	from the fourth quintuple of M_1
(begin, 0, q_2'', b, left)	from the fourth quintuple of M_1
(q_2'', b, halt, X, -)	from the first quintuple of M_1
(begin, E, halt, b, -)	from the third quintuple of M_1.

The first eight quintuples result from adding two quintuples to M_1' for each quintuple in M_1. The final two quintuples result from adding one quintuple to M_1' for each quintuple in M_1 with the initial state as its first element.

Sample input tape for M_1':

... b b b b b b b 0 b . . .
↑

Resulting out put tapes:

... b b b b b b X b b . . .
↑

... b b b b X X X b b ...
↑

... b b X X X X X b b . . .
↑

etc.

APPLICATIONS

Where this investigation was motivated by a problem in a formal system for aesthetics [3], the construction seems widely applicable, e.g. in the theory of automatic programming. If M is a universal Turing machine then M' produces the possible input tapes and (specifications of) Turing machines that could produce a given output tape. In particular, the construction can be used to obtain a **Turing** machine that produces (specifications of) all the possible Turing machines for a particular set of input tape : output tape pairs.

REFERENCES

1. J. McCarthy, "The inversion of functions defined by Turing machines", in C. Shannon and J. McCarthy (eds.), Automata Studies, Princeton University Press, Princeton, New Jersey, 1956.

2. M. Minsky, Computation: Finite and Infinite State Machines, Prentice Hall, Englewood Cliffs, New Jersey, 1967.

3. J. Gips, and G. Stiny, "Aesthetics systems", Computer Science Report No. 337, Stanford University, 1973. Also, submitted for publication.