IR-00021
September 9, 1955

On the Speed of Turing Machines

John McCarthy

Abstract: We give some results on the speed at which Turing machines with various numbers of tapes can simulate other Turing machines.

We consider Turing machines (abbreviated TM's) with varying numbers of tapes of various dimensions. We first consider universal machines for simulating machines with one 1-dimensional tape.

We shall first describe informally a universal machine U_{11} with a single one-dimensional tape for simulating machines with a single one-dimensional tape. Let T be the simulated machine which for the time being we will restrict to use the two symbols 0 and 1. Let T have N(T) states. It is described on the tape of U as follows: Each state q of T is represented as a binary number and corresponding to each state there is an area on the tape. This area has printed on it $\alpha \ldots \beta_i \ldots \gamma \ldots \delta$. Here the dots represent strings of 0's and 1's. The number between the α and the β_i is that of q, the number between β_i and γ gives the new state of T if an 0 is read in state q and the number between γ and δ gives the new state if a 1 is read. The symbol β_i describes the function giving new symbol printed and the direction of motion in terms of the symbol read.

The tape region of U_{11} used to describe T is punctuated with special symbols at the ends. The rest of the tape of U_{11} is used to represent the tape of T.

In order for U_{11} to simulate one step of T it must go through the following procedure starting, let us say, looking at the square of U_{11} corresponding to the state of T. It recognizes this area because the 0 or 1 has been replaced by a special symbol. Next it looks for the succeeding state of T, doing this by comparing the bits of the sequences between various pairs with the $\beta_i \cdots \gamma$ sequence if the symbol read was a zero and with the $\gamma \cdots \delta$ sequence if the symbol read was a 1. In general this requires a trip of half the length of the description of T for each bit of the state number, hence log N(T) trips. After marking the new state area it goes back to the old, erases the mark, carries the symbol read to the β_i , calculates the new symbol and new direction, looks for the square marker, moves right or left and starts the cycle over.

There are three important omissions from the above description. First the states governing searches must have extra bits to determine the direction of search. Second, when a move encounters the area where on T is described it must skim over this area. The third modification is more important.

If U_{11} were as previously described the number of steps of U_{11} required to simulate one step of T would be unbounded since T might be operating on squares an arbitrary distance from the region on which T is described. In order to avoid this, the description of T must be moved whenever the working square gets too far away from it. The way we do this is to designate a region of the tape next to the description of T as a counter and to count it up or down one every time a move is made on T. When the counter reaches too large a positive or negative value the desxription of T is copies on the other side of the scanned square and the old description area is erased.

The Turing machine U_{11} described above will have about 100 states and 30 symbols. For every Turing machine T there is a number k(T) such that U_{11} never takes more than k (T) steps to simulate one step of T. If we consider the operations U_{11} has to perform in order to simulate T and assume that the distance the scanned square is allowed to move before the description is copied is proportional to the length of the description of T arrive at the result that k(T) is in general proportional to $N(T) \log N(T)^2$ where N(T) is the number of internal states of T.

If T has more than two symbols it must be replaced by an equivalent machine using only two symbols before it can be simulated by U_{11}. If T has M(T) symbols the equivalent two symbol machine has $2N(T)\,(2^{p+1} - 1)$ states where p is the smallest integer greater than or equal to log M(T). This equivalent machine makes no more than 3p-2 steps for every step of T. Combining these results we see that U_{11} takes not more than

$$K\,(3p-2)\;2N(T)\;(2^{p+1}-1)\;\log\left\{2N(T)\;(2^{p+1}-1)\right\}$$

steps for every step of T. K is probably about 4.

This indicates that U_{11} takes about 400,000,000 steps to simulate one step of itself.

Let us examine how the speed of simulation can be improved if two tapes are available on the machine doing the simulating. The most natural thing to do in this case is to put the description of T on one tape and use the other working tape to simulate the tape of T. This eliminates the necessity of moving back and forth between the description of T and its working area. The necessity of copying the description of T is also eliminated. This means that most of the time will be spent in finding the area on the description corresponding to the succeeding step of T.

For every large T's this operation too most of the time so that not much improvement is made by only putting the description on another tape except when small Turing machines are being simulated.

Suppose we want to find the number between a α and β_i which is the same as the number between a certain β_i and γ. First we exchange bits between the tapes so that the number between β_i and γ is copied on the working tape and replaced by the same number of bits from that working tape. Then we can search the description tape to find the number we want without having to go back and forth on that tape since the comparison number is at hand on the working tape. After the new state has been found and marked it is necessary to go back to the original state and re-exchange data between the tapes. In this form the time required to simulate a one-step of T is proportional to N(T) log N(T).

If additional tapes are available they can be used to reduce the time required to find the new state by putting the description of T on several tapes. In general this reduction is in proportion to the number of tapes used for the description.

Since the numbering of the states of T is arbitrary they can be arranged on the tapes so as to minimize the time hunting for the next state. In general the best arrangement will not be very much better than the worst but if T has a special structure very large improvements are possible.

If U has a two dimensional tape as well as a one dimensional tape it can simulate 1-dimensional machines with k(T) proportional to $\sqrt{N(T)} \log N(T)$ since the time required to get to the new state will be proportional to $\sqrt{N(T)} \log N(T)$. If an n-dimensional tape is available k(T) is $N(T)^{1/n} \log N(T)$.

In general a universal machine with one more tape than the machine being initiated will function without having to make predictable long journeys over irrelevant data.

John McCarthy