Simulating a nondeterministic Turing machine by a deterministic one

Vassos Hadzilacos

Theorem 3.1: *For every nondeterministic Turing machine (NTM) $M = (Q, \Sigma, \Gamma, \delta, q_0, h_A, h_R)$ there is a deterministic Turing machine (DTM) M' that simulates M; that is, for every input $x \in \Sigma^*$, if M accepts (respectively, rejects or loops on) x then M' accepts (respectively, rejects or loops on) x. Furthermore, if the computation tree of M on x has an accepting path with t moves then M' accepts x in $O(2^{ct})$ moves, for some constant c.*

PROOF. Intuitively, the DTM M' performs a breadth-first search of the computation tree of M on x until it encounters a configuration with the accept state, in which case it accepts, or until all computation paths end with the reject state, in which case it rejects. The construction of M', given M, was shown in lecture (see the document http://www.cs.toronto.edu/~vassos/teaching/c63/handouts/NTM-slides.pdf). Note that this construction is a little different from the one described in your textbook; it is the construction assumed in what follows.

Suppose the computation tree of M on x contains an accepting computation of length t. Without loss of generality we assume that M at least reads its input x (if not, we can modify it so that it does), and therefore $t \geq |x|$. Thus, the length of any configuration reachable from the initial configuration within t moves is at most $t + 1$: t for the maximum number of cells that M can visit in t moves plus one for the state.

Let k be the maximum number of choices that M has from any state; that is,

$$k = \max\{|\delta(q, a)|\colon q \in Q - \{h_A, h_R\} \text{ and } a \in \Gamma\} \leq |Q| \cdot |\Gamma| \cdot 2.$$

Note that this is a constant: It depends only on M and not on the size of the input. The number of configurations of M that M' considers during its breadth-first search exploration of the computation tree of M on x until it discovers the halting configuration at depth t is at most $\Sigma_{i=0}^{t} k^i = O(k^t)$. For each of these configurations, M' performs a number of moves that is proportional to its length, that is, at most $O(t)$ moves: It copies it once from tape 1(the one that contains the queue of configurations) to tape 2 (the work tape), modifies it on tape 2, and then transfers it back to the end of tape 1. Since M' considers $O(k^t)$ configurations before it halts and performs $O(t)$ moves for each of them, it performs $O(t \cdot k^t)$ moves before it accepts x.

Since k is constant, there is some constant d such that $k \leq 2^d$; furthermore $t \leq 2^t$ for every t. Therefore $t \cdot k^t \leq 2^t \cdot 2^{dt} = 2^{ct}$, where $c = d + 1$. So, the total number of moves before M' accepts x is $O(2^{ct})$, for some constant c, as wanted. □

Recall that we can simulate a multi-tape TM by an ordinary (single-tape) TM with a quadratic slow-down: A computation of the multi-tape TM that makes t moves can be simulated by a single-tape TM that makes $O(t^2)$ moves. This is only a polynomial penalty for the simulation. The situation is much worse in the simulation of a NTM by a DTM: Now the penalty is exponential: A t-move computation by the NTM requires $O(2^{ct})$ moves by the DTM.