The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual

Chapter 6, Problem 6.25

First we verify that any problem with class $\mathscr{C}$ many-one reduces (within the appropriate resource bound) to the bounded halting problem for $\mathscr{C}$. Our set is given as $\{(M, x) \mid M \in \mathscr{C}$ and M accepts $x\}$. If some problem Π belongs to the class $\mathscr{C}$, we must have a machine M_Π for it—we have defined all four classes NL, P, NP, and PSPACE, through machines, so that proof of membership in these classes is the demonstration of a suitable machine. Thus, given some instance x of Π, we simply map it into the pair (M_Π, x); this transformation is just an injective embedding that takes linear time and at the very most logarithmic space and clearly is a valid many-one reduction to our set.

Now we need to show that our set belongs to the class $\mathscr{C}$ itself. But that is totally obvious in view of its definition: given the pair (M, x), we simply run M on x, which takes exactly the appropriate amount of resources for the class, and obtain the answer. One subtle point of major importance that we need to point out here is that the definition of the set includes the somewhat cavalier statement $M \in \mathscr{C}$. How do we verify that a machine belongs to the class? Granted that, if the machine does belong to the class, membership of our set to the class is trivial; but what of the machine? Fortunately, we saw, in our development of the hierarchy theorem, that we can use a variety of techniques to bound the running space (by prior marking of the tape) or running time (by keeping a counter on the side) of a machine and that other issues (such as space-bounded machines that never halt) can also be handled. Thus we can take some machine M and make sure, when we run it on x, that it respects the resource bounds for the class, at least for those classes mentioned here.

The key to these syntactic classes is that the classes themselves are defined through easily characterizable machines and so the bounded halting problem is the most natural characterization of the class—although perhaps not a very useful problem for further reductions.