The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 7, Problem 7.5

FP is the class of all functions computable in polynomial time; P-easy problems form the class of all problems that Turing reduce in polynomial time to some problem in P. We need to show containment in each direction.

If f is some function in FP, it is computable in polynomial time; that is, there exists a Turing machine M and a polynomial p such that, on any input x, M computes $f(x)$ and stops after at most $p(|x|)$ steps. In particular, note that the value computed by M on input x uses at most $p(|x|)$ symbols in the tape alphabet and thus cannot exceed $c^{p(|x|)}$ for some positive constant c. We claim that the decision version of f, namely, "does $f(x)$ exceed B," is in P and, moreover, that the computation of f can be Turing-reduced to its decision version. That the decision version is in P is easily seen: we just run machine M on x and compare the output to B, a task that takes at most $O(p(|x|))$ time since we can compare on a symbol by symbol basis. That we can reduce the computation of f to its decision version takes a little more work: we use binary search with B in the range from $-c^{p(|x|)}$ to $c^{p(|x|)}$ to find the value of $f(x)$; each probe in the binary search is a call to the decision version of the problem; at most $\log_2(2c^{p(|x|)})$ such calls will be made, which is $O(p(|x|))$ calls and thus polynomial. Thus any function in FP defines a P-easy problem.

Conversely, let Π be a P-easy problem. Then there exists a Turing reduction from Π to some problem in P; this Turing reduction, along with the algorithm that solves the problem in P, provides a polynomial-time solution for Π, so Π is in FP.