The Theory of Computation
Bernard M.E. Moret
Addison-Wesley-Longman, 1998

Solution Manual
Chapter 6, Problem 6.19

We begin by proving the easy half: if a set is the range of an honest polynomially computable function, then it is in NP. Let S be such a set; that is, there exists some honest polynomially computable function f such that $f(\Sigma^*) = S$. Let the polynomial bound on the length of f be $p_1()$ and let the polynomial bound on the running time of f be $p_2()$, and pick some new polynomial $p()$ that is everywhere larger than both $p_1()$ and $p_2()$. Now, given some element $x \in S$, we know that there exists some string y such that: (i) $f(y) = x$; (ii) $|y| \leq p(|x|)$; and (iii) $f(y)$ is computable in $p(|y|)$ time—which is at most $p(p(|x|))$ time by (ii). There is clear parallel between y and a certificate: like an NP certificate, y depends on its matching x, is concise (bounded by some polynomial in the length of x), and leads to a polynomial-time deterministic computation (here that of $f(y)$). Given an element x of S, the certificate is the string y described above and the checker simply computes $f(y)$ and verifies that $f(y) = x$; thus every element of x has a concise certificate checkable in polynomial time, so that S is in NP.

We now prove the harder half: every set S in NP is the range of an honest polynomially computable function. If a set is in NP, there exists a TM M and a polynomial $p()$ such that, for each $x \in S$, there is a certificate c_x such that $M(x, c_x)$ accepts after at most $p(|x|)$ steps. Using M, $p()$, and the existence of concise certificates, we must show that S is the range of an honest, polynomially computable function f; in other words, $f()$ must produce every element of S and nothing else, and must do so honestly and in polynomial time. In attempting to reverse the reasoning used above, we could define $f(y) = x$ whenever $M(x, y)$ accepts in at most $p(|x|)$ steps and $|y| \leq p(|x|)$, i.e., if y is a concise certificate for x. This approach fails for at least three reasons: (i) it leaves $f()$ ill-defined, because there could be many different $x \in S$ for which the same y is a valid concise certificate (for instance, a truth assignment to n variables is a certificate for a very large collection of possible Boolean formulae); (ii) even if this problem were somehow resolved, it is not clear that $f()$ would generate all elements of S; and (iii) there is no obvious way to compute this function, short of searching through all possible strings x until one is found for which $M(x, y)$ accepts—an exponential-time search. These three observations lead us to a working solution: since all we have time to do is little more than running the checker machine M on one pair of values, we should encode both values into the argument of $f()$—or rather, since the argument is given, we should regard it as the encoding of a pair of values, just as in a dovetailing argument. Let the enumeration of the two indices be given by the dovetailing function $\langle -, - \rangle$; given an argument y for $f()$, we regard it as being produced by the dovetailing function, i.e., we write $y = \langle y_1, y_2 \rangle$. Note that this will automatically enumerate every element of Σ^* in each coordinate, so that we shall not have to worry about missing elements of S.

Let z_0 be some arbitrary element of S (we can always compute one off-line if needed). Given $y = \langle y_1, y_2 \rangle$, we begin by decoding y to recover y_1 and y_2. Next, we check whether $|y_2| \leq p(|y_1|)$; if so, then we run $M(y_1, y_2)$ for at most $p(|y_1|)$ steps. If M accepts, then we set $f(y) = f(\langle y_1, y_2 \rangle) = y_1$—because we have established that y_2 is a concise certificate for y_1 and thus that y_1 is an element of S. If either test fails, we set $f(y) = z_0$.

By construction, it is clear that the range of f is contained in S and that f can be computed in polynomial time (assuming that we can decode the dovetailing pair in polynomial time, which is easy). We are left to verify that every element of S is in the range of $f()$ and that f is honest. But both follow immediately from the hypotheses: since every element x to be produced by f is in $S \in \text{NP}$, it has a concise certificate c_x; let $y = \langle x, c_x \rangle$, then it follows that $f(y) = x$, an honest production of x. (Note that our function is not honest for every single argument: many of the arguments y that lead to $f(y) = z_0$ are such that $|y| > p(|z_0|)$; but that does not matter, since at least one of the arguments produces z_0 honestly.)