COMPUTABILITY:

ANSWERS TO SOME EXERCISES IN CHAPTER 1

1.3.3, P. 21.

1. (a) (If $x = 0$, stop. Else, set $r_1 = 0$. Add 1 to r_1.)

I_1 $J(1, 2, 4)$
I_2 $Z(1)$
I_3 $S(1)$

(b) (Set $r_1 = 0$. Add 1 to r_1 five times.)

I_1 $Z(1)$
I_2 $S(1)$
I_3 $S(1)$
I_4 $S(1)$
I_5 $S(1)$
I_6 $S(1)$

(c) (If $x = y$, go to I_5 for setting $r_1 = 0$. Else, turn r_1 into 1 and stop.

I_1 $J(1, 2, 5)$
I_2 $Z(1)$
I_3 $S(1)$
I_4 $J(1, 1, 6)$
I_5 $Z(1)$

(d) The typical configuration in the machine is $x \quad y \quad k \quad k \quad 0 \quad 0$

I_1 $J(1, 3, 6)$ (if $x = k$, go to I_6 for $r_1 = 0$ and stop)
I_2 $J(2, 4, 8)$ (if $y = k$, go to I_8 followed by I_9 for $r_1 = 1$)
I_3 $S(3)$ ($r_3 := r_3 + 1$)
I_4 $S(4)$ ($r_4 := r_4 + 1$)
I_5 $J(1, 1, 1)$ (return to I_1)
I_6 $Z(1)$ (set $r_1 = 0$)
I_7 $J(1, 1, 10)$ (stop)
I_8 $Z(1)$ (set $r_1 = 0$)
I_9 $S(1)$ (turn $r_1 = 0$ to $r_1 = 1$)

(e) The typical configuration of the machine is $x \quad 3k \quad k \quad 0$

$I_1 \quad J(1,2,7) \quad$ (if $x = 3k$, go to I_7)
$I_2 \quad S(3) \quad (r_3 := r_3 + 1)$
$I_3 \quad S(2)$
$I_4 \quad S(2)$
$I_5 \quad S(2) \quad$ (add 1 to r_2 three times)
$I_6 \quad J(1,1,1) \quad$ (return to the beginning)
$I_6 \quad T(3,1) \quad$ (set $r_1 = k$)

(f) First, we write a subroutine P_1 for computing $f_1(x) = 2x$. The typical configurartion of the machine following this subroutine is $x + k \quad x \quad k \quad 0 \quad \cdots$

$I_1 \quad T(1,2) \quad$ (print $r_2 = x$)
$I_2 \quad J(2,3,6) \quad$ (if $x = k$, stop)
$I_3 \quad S(1) \quad (r_1 := r_1 + 1)$
$I_4 \quad S(3) \quad (r_3 := r_3 + 1)$
$I_5 \quad J(1,1,2) \quad$ (return to I_2 and repeat)

Next we design a subroutine P_2 for computing $f_2(x) = [x/3]$. The concatenation (see the definition on P. 27) of P_1 and P_2 will be a program for computing $f(x) = [2x/3]$. The general configurartion of the machine following the subroutine P_2 is $x \quad k \quad 3k \quad 3k+1 \quad 3k+2 \quad 0 \quad \cdots$

$I_1 \quad J(1,3,15) \quad$ (if $x = 3k$, go to I_{15})
$I_2 \quad J(1,4,15) \quad$ (if $x = 3k+1$, go to I_{15})
$I_3 \quad J(1,5,15) \quad$ (if $x = 3k+2$, go to I_{15})
$I_4 \quad S(2) \quad (r_2 := r_2 + 1)$
$I_5 \quad S(3)$
$I_6 \quad S(3)$
$I_7 \quad S(3) \quad$ (add 1 to r_3, three times)
$I_8 \quad S(4)$
$I_9 \quad S(4)$
$I_{10} \quad S(4) \quad$ (add 1 to r_4, three times)
$I_{11} \quad S(5)$
$I_{12} \quad S(5)$
$I_{13} \quad S(5) \quad$ (add 1 to r_5, three times)
$I_{14} \quad J(1,1,1) \quad$ (go back and repeat)
$I_{15} \quad T(2,1)$

2. Following the program P described by Example 2.1 to compute $f_P^{(2)}(x, y)$, the general configuration of the machine is $x \quad y+k \quad k \quad 0$ It is not hard to see that the answer is

$$f_P^{(2)}(x, y) = \begin{cases} x - y & \text{if } x \geq y; \\ \text{undefined} & \text{otherwise.} \end{cases}$$

3. Suppose that P is a program without any jump instruction. We have to show that there exists some $m \in \mathbf{N}$ such that either $f_P^{(1)}(x) = m$ for all x, or $f_P^{(1)}(x) = x + m$. Denote by $s(P)$ the number of instructions in P. First note that, since P has no jump instruction, any computation following program P will stop after $s(P)$ steps; (the kth step is performed under the instruction I_k). Now $f_P^{(1)}(x)$ is the content of the first register R_1 in the final step of computation. We are going to prove a more general statement:

   (*) the content of any register R_j at any step (say kth step) is either of the form m or $x + m$, where $m = m_{j,k}$ may depend on j and k, but not on x.

   We prove this general statement by induction on k, the number of steps executed. When $k = 0$, the machine is in the initial configuration. We have x in R_1 and 0 in R_j for $j \geq 0$. Thus (*) holds for $k = 0$ with $m_{j,0} = 0$. Now suppose (*) holds for $k = r - 1$ where $r \leq s(P)$. The configuration of the machine right after the rth step is obtained by executing the instruction I_r to the previous configuration, say $\mathcal{C}_{r-1}$, in which the contents of registers are of the form $m_{j,r-1}$ or $x + m_{j,r-1}$, in view of our induction hypothesis. It remains to check to effect of I_r on $\mathcal{C}_{r-1}$, for three types of instructions: $Z(n)$, $S(n)$, $T(m, n)$. This is straightforward.

4. This is because the instruction $T(m, n)$ can be carried out by the following subroutine:

   I_1 $Z(n)$ (set $r_n = 0$)
   I_2 $J(m, n, 5)$ (if $r_m = r_n$, stop)
   I_3 $S(n)$ ($r_n := r_n + 1$)
   I_4 $J(1, 1, 2)$ (return to I_2 and repeat)

1.4.3, P. 23.

1. (a) The function $f(x, y)$ in part (d) of Exercise 1.3.3.1 is just the characteristic function of the predicate '$y < x$'. Switch the letters x and y.

(b) The following program computes the characteristic function of '$x \neq 3'$:

I_1 $S(2)$
I_2 $S(2)$
I_3 $S(2)$ (turn r_2 into 3)
I_4 $J(1, 2, 6)$
I_5 $Z(1)$
I_6 $Z(1)$
I_7 $S(1)$

(c) We create a program to compute the characteristic function of 'x is even' so the the general configeration of the nachine is $x \quad 2k \quad 2k+1 \quad 0$

I_1 $S(3)$
I_2 $J(1, 3, 8)$
I_3 $J(1, 2, 9)$
I_4 $S(2)$
I_5 $S(2)$
I_6 $S(3)$
I_7 $S(3)$
I_8 $Z(1)$
I_9 $Z(1)$
I_{10} $S(1)$

I skip exercises 1.5.2 on p. 24 because they are rather boring.

2.3.4, P. 32.

1. (a) Apply the sucessor function m times to the zero function $\mathbf{0}$, we get the constant function $\mathbf{m}$, which is computable in view of Theorem 2.3.1.

   (b) Let g be the summing function: $g(x, y) = x + y$. Then

$$nx = g(U_1^1(x), g(U_1^1(x), \ldots g(U_1^1(x), g(U_1^1(x), U_1^1(x))) \cdots))$$

   (the number of the letter g on the right hand side is $n-1$), which is computable in view of Theorem 2.3.1. (Alternatively, we may proceed by induction on n, which has a neater presentation.)

2. We use the fact proved in part (a) of the previous exercise: the constant function $\mathbf{m}$ is computable. Now the assertion follows from Theorem 2.3.1 and the identity $h(x) = f(U_1^1(x), \mathbf{m}(x))$.

3. Exercise 1.3.3.1 (a) says that the function

$$f(x,y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$$

is computable. Notice that f is just the characteristic function of the predicate '$x \neq y$'. Hence '$x \neq y'$ is decidable. We can modify the program for computing f (see the solution to Exercise 1.3.3.1 above), with 0 and 1 switched, to show that the predicate $E(x,y) =$ '$x = y'$ is decidable. Clearly

$$c_M(x,y) = c_E(g(U_1^2(x,y)), U_2^2(x,y))$$

which is computable in view of Theorem 2.3.1. Hence M is decidable.

2.4.16, P. 41

1. (b) $[\sqrt{x}]$ by definition is the largest $n \in \mathbf{N}$ such that $n^2 \leq x$. It can also be interpreted as the smallest n such that $x < (n+1)^2$. Notice that: 1. if $n = [\sqrt{x}]$, then $n^2 \leq x < x+1 < (x+1)^2$, or $n < x+1$; 2. $x < (n+1)^2$ iff $x+1 \leq (n+1)^2$, or $(x+1) \dot{-} (n+1)^2 = 0$. Thus we have

$$[\sqrt{x}] = \mu\, y < x+1\, (f(x,y) = 0) \quad \text{where} \quad f(x,y) = (x+1) \dot{-} (y+1)^2 \text{ is computable.}$$

From Theorem 2.4.12 we know that $[\sqrt{x}]$ is computable.

(b) (The expressions $\mathrm{LCM}(x,y)$ and $\mathrm{HCF}(x,y)$ are rather troublesome when $x = 0$ or $y = 0$: our convention is $\mathrm{LCM}(x,t) = 0$ when $x = 0$ or $y = 0$ occurs. From Theorem 2.4.5 part (ℓ) that the remainder function $\mathrm{rm}(x,y)$ and the function $\max(x,y)$ are computable. Now

$$\mathrm{LCM}(x,y) = \mu\, z < xy + 1\ (\max(\mathrm{rm}(x,z),\ \mathrm{rm}(y,z)) = 0)$$

and hence is computable.

(d) $\mathrm{HCF}(x,y) = \mathrm{qt}(\mathrm{LCM}(x,y))$.

(e) Number of prime divisors of $x = \sum_{y<x+1} \mathrm{div}(p_y, x)$.

(f) Define

$$\mathrm{pr}(x,y) = \begin{cases} 1 & \text{if } x \text{ and } y \text{ are relatively prime, i.e. HCF}(x,y)=1; \\ 0 & \text{otherwise.} \end{cases}$$

Since we have $\phi(x) = \sum_{y<x} \mathrm{rp}(x,y)$, it is enough to show that $\mathrm{rp}(x,y)$ is computable. Now $\mathrm{rp}(x,y) = g(\mathrm{HCF}(x,y))$, where $g(x)$ is the function defined by

$$g(x) = \begin{cases} 1 & \text{if } x = 1; \\ 0 & \text{if } x = 0 \text{ or } x > 1. \end{cases}$$

It is enough to check that $g(x)$ is computable. Indeed, $g(x) = \overline{\mathrm{sg}}(x \dot{-} 1) \dot{-} \overline{\mathrm{sg}}(x)$.

2. The function $\pi(x,y)$ in this exercise is instrumental for the proof of part (a) of Theorem 4.1.2. Basic idea behind the function $\pi(x,y)$ is this. Given any positive integer n, we pull out all factors of 2 from n (say r of them) so that we may write $n = 2^r s$. Here s must be an odd number and hence can be written as $2k+1$ for some $k \in \mathbf{N}$. From this explanation it is clear that $\pi \colon \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ is a bijection.

   We know that the functions $f(x) = 2^x$, $f_2(x) = 2x+1$ and $g(x,y) = xy \cdot -1$ are compuatble. Hence so is

$$\pi(x,y) = g(f_1(x), f_2(y)) \equiv g(f_1(U_1^2(x,y)), f_2(U_2^2(x,y))).$$

   Also, we know that the functions $(x)_2$ and $[y/x] \equiv \mathrm{q}(x,y)$ are computable. Thus

$$\pi_1(z) = (z+1)_2 \qquad \text{and} \qquad \pi_2(z) = [(([(z+1)/(z+1)-2]-1)/2]$$

   are also computable.

3. Let $g(x) = 2^{f(x)} 3^{f(x+1)}$. Also let $\eta_1(x) = (x)_2$ and $\eta_2(x) = (x)_3$. We know that $\eta_1(x)$ and $\eta_2(x)$ are computable. Since $f(x) = \eta_1(g(x))$, it is enough to show that $g(x)$ is computable. To this end we use Theorem 2.4.2 and show that $g(x)$ can be computed in a recursive fashion. Indeed, $g(0) = 6$ and

$$\begin{aligned} g(x+1) &= 2^{f(x+1)} 3^{f(x+2)} = 2^{f(x+1)} 3^{f(x)+f(x+1)} \\ &= 2^{\eta_2(g(x))} 3^{\eta_1(g(x))+\eta_2(g(x))} = h(g(x)) \end{aligned}$$

   where $h(z) = 2^{\eta_1(z)} 3^{\eta_2(z)+\eta_2(z)}$ is clearly computable.

4. (a) Let $M(x) =$ 'x is odd$'$ Then $c_M(x) = 1 \cdot -\mathrm{div}(2,x)$, which is computable.

(b) Let $M(x)$ = 'x is a power of prime'; (0 is considered as a power of prime). Recall that the functions p_x and $g(x, y) = (x)_y$ (see Theorem 2.15.(c) and (d)) are computable. Consequently

$$c_M(x) = \sum\nolimits_{y<x} \overline{\mathrm{sg}}(|x - p_y^{g(x,y)}|)$$

is computable.

(c) To facilitate our description, instead we prove that the problem M = 'x is a perfect square' is decidable. We know that the function $f(x) = [\sqrt{x}]$ is computable; see Exercise 4.16.1 (b). Notice that x is a perfect square iff $x = f(x)^2$. Thus we have

$$c_M(x) = \overline{sg}(|x - f(x)^2|)$$

which is clearly computable.

5. See the proof on P. 97 of Cutland's book.

2.5.4, P. 45.

1. Assume that f is a total injective computable function. Then the (partial) function f^{-1} is also computable in view of

$$f^{-1}(x) = \mu\, y\, (|x - f(y)| = 0).$$

2. This is because $f(x) = \mu\, y\, (|p(y) - x|)$.

3. Let $g(x, y) = \mu\, z\, (|zy - x| = 0)$. Then g is computable. Notice that, if $y \neq 0$, or $y = 0$ but $x \neq 0$, we have $g(x, y) = f(x, y)$. The only discrepancy here is $f(0, 0) = 0$ but $g(0, 0)$ is undefined. Let $h(x, y)$ be the function such that $h(0, 0)$ is undefined and $h(x, y) = 0$ otherwise. Then $f(x, y) = g(x, y) + h(x, y)$ and hence is computable.