Exercise sheet 6: Review

Turing machines

1. Consider a k-head Turing machine having a single tape and k heads; more than one head can be on the same cell at a time. At each move, the TM will read the symbols under its heads, and consider its internal state (a unique state for whole machine), then it changes the state, writes a symbol on each cell under a head (if there are more than one head in the same cell, it writes the symbol with only one of them) and moves each head to the left or to the right independently. Prove that the languages accepted by k-head Turing machines are the same languages accepted by ordinary TM's.

2. Consider the numeric functions $f : \mathbb{N}_0^k \to \mathbb{N}_0$.

   - Natural numbers can be represented in unary form: *e.g.* $\bar{0} = 1$, $\bar{1} = 11$, $\bar{2} = 111$, ... Hence we can take the input alphabet of a TM computing numeric functions to be $\Sigma = \{1\}$.
   - Consecutive numbers on the input are separated by a blank space $\Box$.
   - The TM starts its computation with the head placed on the $\Box$ preceding the first number.

   e.g. for $f(2,0,3)$: (head in state q_0)

	↓											
		1	1	1		1		1	1	1	1	

   - When the computation terminates, then input has been replaced on the tape with the result of the function.

   e.g. if $f(2,0,3) = 4$: (head in state q_f)

	↓											
		1	1	1	1	1						

   We make some assumptions to ease composition of TM's:

   - There is a single final state q_f. The TM halts with the head over the $\Box$ just before the solution.
   - The only transition from q_0 is $\delta(q_0, \Box) = (q_i, \Box, R)$.
   - There are no transitions entering q_0 or of the form $\delta(q_f, \Box)$.
   - The computation loops whenever $f(m)$ is undefined.

   This allows us to sequentially compose TM's, as represented by the following diagram

   $$\xrightarrow{(I)} \boxed{M_1} \xrightarrow{(II)} \boxed{M_2} \xrightarrow{(III)}$$

   I Initial state of M_1 and of the combination.

   II Final state of M_1 and initial state of M_2.

   III Final state of M_2 and also of the combination.

   (a) Construct a TM computing the successor function $s(n) = n + 1$.

   (b) Construct a TM computing the zero function $z(X^k) = 0$.

   (c) Construct a TM computing the empty function, *i.e.* the function that is undefined for every $n \in \mathbb{N}_0$.

   (d) Construct a TM computing the projector $u_k^{(n)}(x_1, \ldots, x_k, \ldots, x_n) = x_k$.

   (e) Construct a TM computing the predecesor function $Pd(n) = \begin{cases} 0 & \text{if } n = 0 \\ n-1 & \text{otherwise} \end{cases}$

   (f) Using sequential composition, construct a TM computing the constant function *one*. Tip: $one = \Phi(s^{(1)}, z^{(n)})$.

Primitive recursive functions

3. Let $\lessdot$ be a primitive recursive relation. Show that the following functions are primitive recursive.

   (a) $f_1(x, y_0, y)$ = the first value z in $[y_0, y]$ for which $x \lessdot z$.

   (b) $f_2(x, y)$ = the second value z in $[0, y]$ for which $x \lessdot z$.

   (c) $f_3(x, y)$ = the largest value z in $[0, y]$ for which $x \lessdot z$.

   If there is not value z in the range such that $x \lessdot z$, then f_i is $y + 1$.

4. Let f and g be primitive recursive functions. Show that the following function is also primitive recursive:

$$h(x) = \begin{cases} 1 & \text{if } f(i) > g(j), \text{ for all } 0 \leq i \leq x \text{ and } 0 \leq j \leq x \\ 0 & \text{otherwise} \end{cases}$$

5. Give primitive recursive definitions for the following functions

   (a) $\text{half}(x) = \lfloor \frac{x}{2} \rfloor$.

   (b) $\min(x, y)$.

   (c) $\min^n(x_1, \ldots, x_n)$ for all $n \geq 2$.

   (d) $\max(x, y)$.

   (e) $rem(a, b)$ = remainder of the division of a by b.

   (f) $quo(a, b)$ = quotient of the division of a by b, with $quo(a, 0) = 0$.

6. Show that the following function is primitive recursive:

$$f(0) = 0, \qquad f(1) = 1, \qquad f(2) = 2^{2^2}, \qquad f(3) = 3^{3^{3^3}}, \qquad \ldots \qquad f(n) = n^{n^{\cdot^{\cdot^{\cdot^{n}}}}} \quad (n \text{ times})$$

Recursive functions

7. Show that the following are recursive functions

   (a) $f(x, y) = \lfloor \log_x y \rfloor$

   (b) $g(x, y) = \lfloor \log_x y \rfloor + \lceil \sqrt[y]{x} \rceil$

   (c) $g(x, y) = \lceil \sqrt[4]{x + y} \rceil$

Mixed

8. Show that the following function is recursive primitive.

$$f(x, y) = \begin{cases} x + y & \text{if } x \text{ is an even number, multiple of 3.} \\ x \dot{-} y & \text{if } x \text{ is an odd number, multiple of 3.} \\ x & \text{otherwise.} \end{cases}$$

9. Consider the following function:

$$f(x, y) = \frac{x^2}{y}$$

   Is it recursive primitive? In such case, write it as so, in other case, write it as recursive function (if possible). Justify in any case.