CSL705: Theory of Computation

Tutorial sheet: Primitive Recursion & Universality

1. Prove that the following functions on the naturals are primitive recursive.

   prod . The binary product of two numbers

   texp . The binary <u>total</u> exponentiation function (assuming $0^0 = 1$).

   tpred . The <u>total</u> predecessor function (assuming that the predecessor of 0 is 0 itself)

   monus . The binary <u>total</u> subtraction function (denoted by $\overset{t}{-}$) where $x \overset{t}{-} y = 0$ whenever $x < y$, otherwise it is just like subtraction.

   sg . The unary sign function which yields 1 when the argument is positive and 0 otherwise.

   $\overline{\text{sg}}$. The unary "is-zero" predicate.

   diff . The binary function $|x - y|$.

   ! . The usual (unary) factorial function

   min . The binary function yielding the minimum of the two arguments.

   max . The binary function yielding the maximum of the two arguments.

   trem . The binary <u>total</u> remainder (assume division by 0 yields the dividend as the remainder).

   tquot . The binary <u>total</u> quotient function (assume division by 0 yields 0 as the quotient)[1]

   tdivides . The <u>total</u> binary predicate "divides" assuming $0|0$ but for all $y \neq 0$, $0 \nmid y$.

2. Let $f : \mathbb{N} \to \mathbb{N}$ be a primitive recursive function.
   (a) Prove that $h_1 : \mathbb{N} \to \mathbb{N}$ defined as $h_1(x) = f^{2n}(x)$ is also primitive recursive.
   (b) Prove that $h_2 : \mathbb{N} \to \mathbb{N}$ defined as $h_2(x) = f^{2^n}(x)$ is also primitive recursive.
   (c) Let $g : \mathbb{N} \to \mathbb{N}$ be any primitive recursive function. Is $h : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined by $h(n, x) = f^{g(n)}(x)$ also primitive recursive? If so prove that it is primitive recursive and if not give an example to show that it cannot in general be primitive recursive.

3. Euler introduced an important number-theoretic function $\varphi : \mathbb{N} \to \mathbb{N}$ which may be defined as follows.
$$\varphi(n) = \begin{cases} n & \text{if } n \leq 1 \\ k & \text{if } n > 1 \text{ and } k = |\{i \mid 0 < i < n, gcd(i, n) = 1\}| \end{cases}$$
   Prove that Euler's φ function is primitive recursive. *You may define it in terms of the primitive recursive functions already defined in the notes. But you have to define it so that it conforms to the definition of a function being primitive recursive.*

4. Prove that the function $prime(k)$ which computes the k-prime p_k for any positive k is primitive recursive (assume $p_0 = 1$).

5. Prove that the predicate $isprime(n)$ which determines whether a given number is prime, is primitive recursive.

6. Prove that the predicate $isSquareFree : \mathbb{N} \to \{0, 1\}$ (which yields 1 if the number has no divisor that is a perfect square and 0 otherwise) is primitive recursive.

7. Universality makes multiprocessing feasible. Consider the most primitive form of multiprocessing where two independent processes are executed concurrently on the same CPU. Let P_1 and P_2 be URM programs.
   (a) Design a computable function *interleave* which ensures that starting with the first instruction of P_1, instructions from P_1 and P_2 are executed alternately as long as neither P_1 nor P_2 has terminated. As soon as one of them terminates, the machine executes only the other. Assume that if and when P_i $(i = 1, 2)$ terminates, its output is stored in register R_i.
   (b) Prove that *interleave* is primitive recursive.

---
[1] [illegible]