THEORY OF COMPUTATION
Problem session - 7

Prof. Dan A. Simovici

UMB

Problem 1: Two numbers are relatively prime if they have no common factor except 1. Define the predicate $R(x, y)$ as

$$R(x, y) = \begin{cases} 1 & \text{if } x, y \text{ are relatively prime,} \\ 0 & \text{otherwise.} \end{cases}$$

Prove that R is primitive recursive without using quotient, remainder, or the gcd function

Problem 2: Let g be a primitive recursive function. Denote by g^k the function defined as $g^k(x) = g(\cdots(g(x))\cdots)$ is primitive recursive and let $h(x, y) = g^y(x)$. Prove that h is primitive recursive.

Problem 3: Let $f(x, 0) = g(x)$ and $f(x, y + 1) = f(f(x, y), y)$.
Show that if g is primitive recursive, then so is f.

Problem 4: Let $g(x, y)$ be a function. Suppose that f is a function such that $f(n) = g(n, [f(0), f(1), \ldots, f(n-1)])$ for all n. This is the course-of-value recursive definition of f. Prove that if g is a primitive recursive function, then so is f.

Problem 5: Let h be a function defined as

$$\begin{aligned} h(0) &= 3, \\ h(x+1) &= \sum_{t=0}^{x} h(t). \end{aligned}$$

Prove that h is primitive recursive.

Solution for Problem 1: If x, y are not relatively prime they have a common factor that is greater than 1, that is, there exists $t > 1$ such that $t|x$ and $t|y$. Therefore, $R(x, y)$ is $\sim (\exists t)_{\leqslant x}((t > 1)\&(t|x)\&(t|y))$.

Solution for Problem 2: The recursive definition is

$$\begin{aligned} h(x,0) &= g^0(x) = x, \\ h(x,y+1) &= f(y,h(x,y),x), \end{aligned}$$

where $f(x_1,x_2,x_3) = g(x_2)$. The function f is primitive recursive because $f(x_1,x_2,x_3) = g(u_2^3(x_1,x_2,x_3))$, so the above definition of h shows its primitive recursiveness.

Solution for Problem 3: We prove first that $f(x,y) = g^{2^y}(x)$.
For $y = 0$ and for every x, $g^{2^0}(x) = g(x)$, which is $f(x,0)$ by definition.
Suppose that for $f(x,y) = g^{2^y}(x)$. Then,

$$\begin{aligned} f(x, y+1) &= f(f(x,y), y) \\ & \quad \text{(by the definition of } f\text{)} \\ &= f(g^{2^y}(x), y) \\ & \quad \text{(by inductive hypothesis)} \\ &= g^{2^y}(g^{2^y}(x)) = g^{2^{(y+1)}}(x), \end{aligned}$$

which proves the above equality.
Now, x^y is primitive recursive and $g^y(x)$ is primitive recursive, by Problem 2, which implies that f is primitive recursive.

Solution for Problem 4: Let $\tilde{f}$ be defined as

$$\begin{aligned}\tilde{f}(0) &= 1,\\ \tilde{f}(n) &= [f(0), f(1), \ldots, f(n-1)] \text{ if } n \neq 0.\end{aligned}$$

Observe that $f(n) = (\tilde{f}(n+1))_{n+1}$. Thus, if we manage to prove that $\tilde{f}$ is primitive recursive, the primitive recursiveness of f would follow.

Solution for Problem 4 cont'd:

Note that $f(n) = g(n, \tilde{f}(n))$ and

$$\begin{aligned}
\tilde{f}(n+1) &= \tilde{f}(n) \cdot p_{n+1}^{f(n)} \\
&= \tilde{f}(n) \cdot p_{n+1}^{f(n)} \\
&= \tilde{f}(n) \cdot p_{n+1}^{g(n,\tilde{f}(n))}.
\end{aligned}$$

Let $U(n, y)$ be the function defined as $U(n, y) = y \cdot p_{n+1}^{g(n,y)}$. It is clear that U is primitive recursive because all components and operations of $U(n, y)$ are primitive recursive. Since $\tilde{f}(n+1) = U(n, \tilde{f}(n))$, this means that $\tilde{f}$ is primitive recursive, and this implies the primitive recursiveness of f, as we have seen above.

Solution for Problem 5: Let h be a function defined as

$$\begin{aligned} h(0) &= 3, \\ h(x+1) &= \sum_{t=0}^{x} h(t). \end{aligned}$$

Prove that h is primitive recursive.

We show that h can be built through course-of-values recursion, that is, there is a primitive recursive function g such that

$$h(x+1) = g(x, [h(0), h(1), \ldots, h(x)])$$

for all x. We need g to satisfy the equality

$$g(x, [h(0), \ldots, h(x)]) = \sum_{t=0}^{x} h(t).$$

For this, it suffices to define

$$g(x, y) = \sum_{i=1}^{\mathrm{Lt}(y)+1} (y)_{i-1},$$

which is clearly primitive recursive. Thus, h is primitive recursive.