CISC 401/601
Midterm Exam

Answers

1. (5 pts) Let $f$ be a unary one-to-one, onto, and computable function. Show that $f^{-1}$ is also computable.

**Ans.**
Since there exists a corresponding $L$ program for the computable function $f$, the following program is clearly a valid $L$ program.

[A] $Z_2 \leftarrow f(Z_1)$
$Z_1 \leftarrow Z_1 + 1$
IF $Z_2 \neq X$ GOTO A
$Y \leftarrow Z_1 - 1$

Since $f$ is one-to-one and onto, for all $y$ there exists one and only one $x$ such that $f^{-1}(y) = x$, or $f(x) = y$. Therefore, the program above halts on any input and it implements $f^{-1}$. Thus, $f^{-1}$ is computable.

2. (5 pts) Let $f(n)$ be the number of prime numbers $\leq n$. Show $f$ is a primitive recursive function.

**Ans.**
$f$ is defined as a composition of primitive recursive functions as follows:

$$f(x) = \sum_{i=0}^{x} \text{Prime}(i)$$

Therefore, $f$ is also a primitive recursive function.

3. (10 pts) Let $g$ be a primitive recursive function. Let $f(n) = g(n, f'(n))$ where $f'(0) = 1$ and $f'(n) = [f(0), \ldots, f(n-1)]$ when $n > 0$. Show that $f$ is a primitive recursive function.

**Ans.**
First, show that $f'$ is primitive recursive.
The recursion equations are
\[ f'(0) = 1, \]
\[ f'(n + 1) = h(n, f'(n)) \]

and

\[ h(x_1, x_2) = x_2 \cdot f_{x_1+1}^{p_{x_1} x_2}. \]

Clearly, \( f' \) is primitive recursive, and \( f(n) \), a composition of primitive recursive functions, is also primitive recursive.

4. (10 pts) Let \( f \) be computable and also be a strictly increasing function. Let \( B \) be the range of \( f \). Show that \( B \) is recursive.

**Ans.**

Show that the characteristic function of \( B \), say \( P_B \), is computable. \(^1\)

Given a strictly increasing function \( f \), note that \( f(y) \geq y \) for all \( y \). \(^2\)

Also, for all \( x > y \), \( f(x) \neq y \) because \( f(x) > f(y) \geq y \), i.e., if there exists \( z \) such that \( f(z) = y \), then \( z \leq y \).

In other words, given \( y \), it is sufficient to check \( y + 1 \) cases at most to determine if \( y \) is in the range of \( f \) or not, namely to check \( f(0) = y \), \( f(1) = y \), \( \ldots \) up to \( f(y) = y \) and see if there is \( t \) such that \( f(t) = y \). Thus,

\[ P_B(y) = (\exists t \leq y)[f(t) = y]. \]

Clearly, \( P_B \) is computable.

5. a. (5 pts) Let \( g \) and \( h \) be partially computable functions of one argument. Show that there is a partially computable function, \( f \), such that \( f(x) \downarrow \) for precisely those values of \( x \) for which \( g(x) \downarrow \) or \( h(x) \downarrow \) and such that when \( f(x) \downarrow \), \( f(x) = g(x) \) or \( f(x) = h(x) \).

**Ans.**

There exist \( \mathcal{L} \) programs, say \( G \) and \( H \), for partially computable functions \( g \) and \( h \), respectively. Let \( p_g = \#(G) \) and \( p_h = \#(H) \).

\(^1\) “... to say that set \( B \) is computable or recursive is just to say that \( P(x_1, x_2, \ldots, x_m) \) is a computable function.” (text p78)

\(^2\) We can show this by mathematical induction.
Define a $\mathcal{L}$ program $F$ as follows: $^3$

\begin{align*}
[A] & \quad Z_1 = p_y \\
& \quad \text{IF STP}(X, Z_1, Z_2) \text{ GOTO B} \\
& \quad Z_1 = pn \\
& \quad \text{IF STP}(X, Z_1, Z_2) \text{ GOTO B} \\
& \quad Z_2 \leftarrow Z_2 + 1 \\
& \quad \text{GOTO A} \\
[B] & \quad Y \leftarrow \Phi^{(1)}(X, Z_1)
\end{align*}

Clearly, $F$ is a valid $\mathcal{L}$ program.
Moreover, $\Psi_F^{(1)}$ is equivalent to $f$, and, thus, the partially computable function $f$ exists.

\textbf{b.} (5 pts) Can $f$ be found if in addition we require that $f(x) = g(x)$ whenever $g(x) \downarrow$? Justify your answer.

\textbf{Ans.}
No such $f$ can be found.

Prove it by contradiction.
Define $g$ and $h$ as follows:

\[ g(x) = \begin{cases} 
1 & \text{if } \Phi(x, x) \downarrow \\
\uparrow & \text{otherwise}
\end{cases} \]

\[ h(x) = 0 \]

Clearly, $g$ and $h$ are partially computable. $^4$

Suppose there exists $f$.
Clearly, for all $x$, $f(x) \downarrow \in \{0, 1\}$ because $h(x) \downarrow \in \{0\}$ for all $x$, and $g(x) \downarrow \in \{1\}$ for some $x$. Moreover,

\[ f(x) \downarrow = h(x) \downarrow \Leftrightarrow g(x) \uparrow. \]

Therefore,

\[ [f(x) = h(x)] \Leftrightarrow \text{Halt}(x, p_y) \]

where $p_y$ is defined in (a).

---

$^3$ \textit{Dovetailing} (text p80)
$^4$ $g$ is the same as $H_1$ in problem 1 in homework 3. $h$ is obviously computable.
Thus, if \( f \) exists, \([f(x) = h(x)]\) is a computable predicate because \( f \) and \( h \) are both computable, and so is \( \text{Halt}(x, p_y) \). \( \text{Halt}(x, p_y) \), however, is shown to be \textit{not} computable in homework 3 problem 1 (b).

Therefore, we conclude that there is no such \( f \).