CISC 401/601
Homework 1

Answers

1. a. Write a program to compute the predicate $\geq$.

[A] IF $X_2 \neq 0$ GOTO B
    Y ← Y + 1
    GOTO E
[B] IF $X_1 \neq 0$ GOTO C
    GOTO E
[C] $X_1 \leftarrow X_1 - 1$
    $X_2 \leftarrow X_2 - 1$
    GOTO A

b. Write a program to compute the equality predicate.

Note: if $x_1 \geq x_2 \land x_2 \geq x_1$, then $x_1 = x_2$.

    IF $X_1 \geq X_2$ GOTO A
    GOTO E
[A] IF $X_2 \geq X_1$ GOTO B
    GOTO E
[B] Y ← Y + 1

c. Write a program that computes the "is-even" predicate.

[A] IF $Z = X_1$ GOTO B    // $z = 2 \ast n$ for each $n \geq 0$
    IF $Z > X_1$ GOTO E
    $z \leftarrow z + 1$
    $Z \leftarrow Z + 1$
    GOTO A
[B] Y ← Y + 1

2. Write a program to compute $f(x) = n$, where $n$ is the greatest number such that $n^2 \leq x$ (Ch.2, Sec. 2, Ex. 6).

Note: multiplication, $\times$, is a macro defined.

[A] $Z \leftarrow Z + 1$
    IF $X \geq Z \times Z$ GOTO B
    GOTO E
[B] Y ← Y + 1
    GOTO A
3. Ch.2, Sec. 4, Ex. 6.

(a) For every number \( k \geq 0 \), let \( f_k \) be the constant function \( f_k(x) = k \). Show that for every \( k \), \( f_k \) is computable.

For any \( k \geq 0 \), we can write a following program \( P_k \).

\[
\begin{align*}
Y & \leftarrow Y + 1 \\
Y & \leftarrow Y + 1 \\
& \quad \vdots \\
Y & \leftarrow Y + 1
\end{align*}
\]

\( k \) instructions

Obviously, \( \psi^{(1)}_P(x) = k \) for any \( x \geq 0 \), and, thus, \( \psi^{(1)}_P = f_k \). Therefore, for every \( k \), \( f_k \) is computable.

(b) Show by induction on the length of programs that if the length of a straightline program \( P \) is \( k \), then \( \psi^{(1)}_P(x) \leq k \) for all \( x \).

**Basis:** For \( k = 0 \), a program of length 0 is an empty program that always outputs 0.

Clearly, in this case, \( \psi^{(1)}_P(x) \leq 0 \) for all \( x \), where \( P \) is the empty program.

**Hypothesis:** Suppose, for \( k = n \), any straightline program that has a length \( n \) satisfies the claim.

**Induction Step:** Given any straightline program that has length \( n + 1 \), we could see it as a straightline program of length \( n \) followed by one extra instruction at the end. By the hypothesis, immediately before that instruction at the end is executed, the value of \( Y \) is known to be at most \( n \). As a straightline program, the last instruction cannot be jump. So, it is executed to add at most one to \( Y \), and the program halts. Therefore, the output can be \( n + 1 \), at most, and the claim is also satisfied in this case.

By induction, we conclude that the claim is satisfied for every \( k \geq 0 \).

(c) Show that, if \( P \) is a straightline program that computes \( f_k \), then the length of \( P \) is at least \( k \).

Suppose there is a straightline program that computes \( f_k \), but its length, say \( l \), is less than \( k \).
Then, from (b), we know that the output of such program can be at most \( l \) which is less than \( k \). This, however, contradicts to our assumption that the program implements \( f_k \), i.e., it outputs \( k \) for any \( x \). Therefore, we conclude there is no such straightline program.

Knowing also that there is a straightline program of length \( k \) that implements \( f_k \), e.g., (a), we conclude that, if \( P \) is a straightline program that computes \( f_k \), the length of \( P \) is at least \( k \).

(d) Show that no straightline \( \mathcal{L} \) program computes the function \( f(x) = x + 1 \). Conclude that the class of functions computable by straightline \( \mathcal{L} \) is contained in but is not equal to the class of computable functions.

Suppose there is a straightline program, say program \( Q \), that computes the function \( f(x) = x + 1 \), and let \( l \) be the length of \( Q \).

Then, \( \psi_Q(l) \) would be \( l + 1 \) by the definition of \( Q \). This result, however, contradicts to (b) that tells us \( \psi_Q(l) \leq l \) because \( Q \) is a straightline program of length \( l \).

Therefore, we know that there is no straightline program that computes the function \( f(x) = x + 1 \).

Since, obviously, there is a non-straightline program that can compute the function \( f(x) = x + 1 \), and a set of straightline programs is a subset of all (non-restricted) programs in the system, we conclude that the class of functions computable by straightline \( P \) is contained in but is not equal to the class of computable functions.

4. Let \( P(x) \) be a computable predicate. Show that

\[
EX_P(r) = \begin{cases} 
1 & \text{if there are at least } r \text{ numbers } n \text{ such that } P(n) = 1 \\
\uparrow & \text{otherwise}
\end{cases}
\]

is partially computable.

(Ch.2, Sec. 5, Ex. 6).

Notice that, since \( P \) is computable, there is a program that implements \( P \). Using such program, a program for \( EX_P \) is written as follows:
[A] IF $Z_3 = X$ GOTO C // $z_3$ records how many $n$’s we find so far
  $Z_2 \leftarrow P( Z_1 )$
  $Z_1 \leftarrow Z_1 + 1$
  IF $Z_2 = 1$ GOTO B
  GOTO A
[B] $Z_3 \leftarrow Z_3 + 1$
  GOTO A
[C] $Y \leftarrow Y + 1$

Since there is a program implementing $EX_p$, $EX_p$ is partially computable.