CISC 401/601
Homework 4

Answers

1. Let $A = \{ x \in N \mid \Phi_x(x) \downarrow \text{ and } \Phi_x(x) > x \}$. (Q3, p94)

   a. Show that $A$ is r.e.

   \textbf{Ans.}  
   Show that there is a partially computable function $g$ such that $A = \{ x \in N \mid g(x) \downarrow \}$. 

   \begin{align*}
   Z & \leftarrow \Phi( X, X ) \\
   [A] & \text{ IF } Z \leq X \text{ GOTO } A
   \end{align*}

   Clearly, the program above implements $g$, and it is a valid $L$ program, since all macros can be expanded.

   b. Show by diagonalization that $A$ is not recursive.

   \textbf{Ans.}  
   Suppose $A$ is recursive. Then, there is a computable predicate $P_A$ such that $A = \{ x \in N \mid P_A(x) \}$.  

   Given such predicate, consider the following function. 

   \[ h(x) = \begin{cases} 
   \uparrow & \text{if } P_A(x) \\
   x + 1 & \text{otherwise} 
   \end{cases} \]

   Clearly, $h$ is partially computable, and there is a corresponding program number $p \in N$ such that $\Phi_p(x)$ is equivalent to $h(x)$.

   Note that $p$ is either in $A$ or not in $A$.  

   If $p$ is in $A$, $h(p) \uparrow$ by $h$'s definition. It, however, means $\Phi_p(p) \uparrow$, and $p$ should not be in $A$. Therefore, there is a contradiction.

   If $p$ is not in $A$, $h(p) = p + 1$. It, however, implies $\Phi_p(p) \downarrow$ and $\Phi_p(p) > p$, and $p$ should be in $A$. Therefore, there is a contradiction again.

   Hence $P_A$ would not be a computable predicate, and $A$ is not recursive.

   \textit{Alternatively}, we can show that $A$ is not recursive as follows.  

   Knowing that $A$ is r.e., show that $\overline{A}$ is \textit{not} r.e.

Suppose $\overline{A}$ is r.e. Then, there is a partially computable function $g$ such that $\overline{A} = \{ x \in N \mid g(x) \downarrow \}$.

Given $g$, consider the following function.

$$h(x) = \begin{cases} x + 1 & \text{if } g(x) \downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Clearly, $h$ is partially computable, and there is a corresponding program number, say $p$, in $N$. We show that $p$ can be in neither $\overline{A}$ nor $A$ to derive a contradiction.

First, note that $h$ is actually defined as follows.

$$h(x) = \begin{cases} x + 1 & \text{if } x \in \overline{A} \\ \uparrow & \text{if } x \in A \end{cases}$$

Then, consider the following two cases.

Case 1: $p \in \overline{A}$

If $p \in \overline{A}$, then $h(p) = p + 1$, and, that is, $\Phi_p(p) \downarrow > p$. Thus, $p$ should be in $A$, not in $\overline{A}$.

Case 2: $p \in A$

If $p \in A$, then $h(p) \uparrow$, or $\Phi_p(p) \uparrow$. Thus, $p$ should be in $\overline{A}$, not in $A$.

Since $p$ can be in neither $\overline{A}$ nor $A$, we conclude $p$ does not exist.

Therefore, $\overline{A}$ is not r.e., and, thus, $A$ is not recursive.

2. For every number $n$, let $A_n = \{ x \mid n \in W_x \}$. (Q9, p94)

(a) Show that $A_i$ is r.e. but not recursive, for all $i$.

**Ans.**

- First, show that $A_i$ is r.e.

$A_i$ is r.e., if there is a partially computable function $h$ such that $A_i = \{ x \mid h(i) \downarrow \}$, or equivalently, there is a $\mathcal{L}$ program $P$ such that $A_i = \{ x \mid \Phi_P(i) \downarrow \}$. Consider the following program.
1. \[ Z_1 \leftarrow i \quad /\ i \text{ is a constant} \]
2. \[ Z_2 \leftarrow \Phi( Z_1, x ) \]

The program above is a valid \( L \) program \(^1\), and clearly the program halts if and only if \( \Phi(i, x) \downarrow \), or \( i \in W_i \).
Therefore, we conclude that the program above witnesses \( A_i \) is r.e.

Alternatively, we can show that \( A \) is r.e. as follows.
Knowing that \( K \) is r.e., \( A_i \) is r.e. if \( A_i \leq_m K \) by Theorem 6.2 on p91.

Consider the following program.
\[
Z \leftarrow i
\quad \Phi( Z, x_2 )
\]
Obviously, the program above can be a valid \( L \) program, and there is a corresponding program number, say \( p_i \).
Note that, regardless of \( x_1 \),
\[
\Phi(i, x_2) \downarrow \iff \Phi^{[2]}(x_1, x_2, p_i) \downarrow.
\]
By S-m-n,
\[
\Phi(i, x_2) \downarrow \iff \Phi(x_1, s_1^i(x_2, p_i)) \downarrow
\]
Let \( s_1^i(x, p_i) \) be \( f_i(x) \), which is known to be computable.
Then, for all \( x_2 \),
\[
\Phi(i, x_2) \downarrow \Rightarrow \text{for all } x_1, \Phi(x_1, f_i(x_2)) \downarrow \Rightarrow \Phi(f_i(x_2), f_i(x_2)) \downarrow
\Rightarrow f_i(x_2) \in K
\]
Also, for all \( x_2 \),
\[
\Phi(i, x_2) \uparrow \Rightarrow \text{for all } x_1, \Phi(x_1, f_i(x_2)) \uparrow \Rightarrow \Phi(f_i(x_2), f_i(x_2)) \uparrow
\Rightarrow f_i(x_2) \notin K
\]
Therefore,
\[
\Phi(i, x_2) \downarrow \iff f_i(x_2) \in K
\]
and, thus,
\[ A_i \leq_m K. \]

- Next, show that \( A_i \) is not recursive by showing that \( K \leq_m A_i \).

Consider the following program.

\(^1\)Clearly, the line 1 is a proper macro in \( L \) program. Also the line 2 is a proper macro because, knowing that \( \Phi(x_1, x_2) \) is partially computable by Universality Theorem on p70, there is a corresponding \( L \) program for the function. By replacing variable names within that program, clearly the line 2 can be expanded as a proper \( L \) program.
\( Z \leftarrow \Phi( x_2 , x_2 ) \)

Obviously, the program above can be a valid \( \mathcal{L} \) program, and there is a corresponding program number, say \( q \).

Note that, regardless of \( x_1 \),
\[
x_2 \in K \iff \Phi^{(2)}(x_1, x_2, q) \downarrow .
\]

By S-m-n,
\[
x_2 \in K \iff \Phi(x_1, s_1(x_2, q)) \downarrow
\]

Let \( s_1(x, q) \) be \( g(x) \), which is known to be computable.

Then, for all \( x_1 \) including \( i \),
\[
x_2 \in K \iff \Phi(x_1, g(x_2)) \downarrow .
\]

Therefore,
\[
x_2 \in K \iff g(x_2) \in A_i
\]

As a whole,
\[
K \leq_m A_i.
\]

(b) Show that \( A_i \equiv_m A_j \) for all \( i, j \).

\textbf{Ans.}

Knowing that \( A_i \) is r.e. for any \( i \) from (a) and that \( K \) is m-complete by \textbf{Theorem 6.5.} on p93, \( A_i \leq_m K \) for any \( i \). Also from (a), \( K \leq_m A_j \) for any \( j \).

Thus, by the transitivity of \( \leq_m \) found in \textbf{Theorem 6.3.} on p92, \( A_i \leq_m A_j \) holds for any \( i \) and \( j \). Since the relation holds for any \( i \) and \( j \), \( A_j \leq_m A_i \) also holds.

Therefore, \( A_i \equiv_m A_j \) for all \( i \) and \( j \).

\textit{Alternatively}, we can also show that there is a computable function \( f' \) such that \( f'(p) \in A_j \iff p \in A_i \) as follows.

Without loss of generality, we assume \( j \geq i \) in the following.

Given any \( p \in N \), which may or may not be in \( A_i \), we can construct a \( \mathcal{L} \) program as follows.

\[
\begin{array}{l|l}
1 & X \leftarrow X - 1 \\
2 & X \leftarrow X - 1 \\
\vdots & \\
\vdots & \\
j - i & X \leftarrow X - 1 \\
\end{array}
\]
Note that, given $i, j$ and $p$, the program above is a valid $L$ program, and there is a corresponding number for it, say $p'$. Having $i$ and $j$ as constants, $p'$ is a function of $p$. Moreover, considering the coding of $L$ program by numbers, it is primitive recursive. Call the function $f'$.

Observing the program above, for any $p$, $\Phi(x, f'(p))$ is equivalent to $\Phi(x - (j - i), p)$. Thus, $\Phi(j, f'(p))$ is equivalent to $\Phi(i, p)$. Hence, for any $p$,

$$f'(p) \in A_j \iff p \in A_i$$

Therefore, $A_i \leq_m A_j$.

3. Let $\text{INF} = \{ x \in N \mid W_x \text{ is infinite} \}$. Show that $\text{INF} \equiv_m \text{TOT}$. (Q12, p95)

**Ans.**
First, show $\text{TOT} \leq_m \text{INF}$.
Consider the following program.

[A]  
\[
\begin{align*}
\Phi( & Z, X_2) \\
& Z \leftarrow Z + 1 \\
& \text{IF } Z \leq X_1 \text{ GOTO A}
\end{align*}
\]

Obviously, the program above can be a valid $L$ program, and there is a corresponding program number, say $p$.
Moreover, by s-m-n, $\Phi^{(2)}(x_1, x_2, p) = \Phi(x_1, s_1(x_2, p))$, and let $f(x)$ be $s_1(x, p)$, which is computable.

Show that $x_2 \in \text{TOT} \iff f(x_2) \in \text{INF}$.
- Show $x_2 \in \text{TOT} \Rightarrow f(x_2) \in \text{INF}$.
  - If $x_2 \in \text{TOT}$, for all $x_1$, $\Phi^{(2)}(x_1, x_2, p) \downarrow$, or $\Phi(x_1, f(x_2)) \downarrow$.
  - Show $x_2 \in \text{TOT} \Leftarrow f(x_2) \in \text{INF}$.
    - If $x_2 \not\in \text{TOT}$, then there exists $y_0$ such that, for all $y < y_0$, $\Phi(y, x_2) \downarrow$, but $\Phi(y_0, x_2) \uparrow$. Thus, for all $y < y_0$, $\Phi(y, f(x_2)) \downarrow$.
    - But, for any $y \geq y_0$, $\Phi(y, f(x_2)) \uparrow$ because $Z$ in the program above will reach $y_0$ before it can halt, and, for $Z = y_0$, $\Phi(Z, x_2) \uparrow$, i.e., the program stuck at that point.
    - Since $\Phi(y, f(x_2)) \uparrow$ for all $y \geq y_0$, if $x_2 \not\in \text{TOT}$, $f(x_2) \not\in \text{INF}$.

Therefore, we conclude $\text{TOT} \leq_m \text{INF}$.  

Next, we show $\text{INF} \leq_m \text{TOT}$.
Consider the following program.

\[
\begin{align*}
Z & \leftarrow \langle x_1, 0 \rangle \\
[A] & \quad Z \leftarrow Z + 1 \\
& \quad \text{IF } \text{STP}(l(Z), x_2, r(Z)) \neq 1 \text{ GOTO A}
\end{align*}
\]

Obviously, the program above can be a valid $\mathcal{L}$ program, and there is a corresponding program number, say $q$.

By s-m-n, $\Phi^q(x_1, x_2, q) = \Phi(x_1, s_1(x_2, q))$, and let $g(x)$ be $s_1(x, q)$, which is computable.

Show that $x_2 \in \text{INF} \iff f(x_2) \in \text{TOT}$.

- Show $x_2 \in \text{INF} \Rightarrow f(x_2) \in \text{TOT}$.  
  
  If $x_2 \in \text{INF}$, for any $x_1$, there exists $y \geq x_1$ such that $\Phi(y, x_2) \downarrow$.
  Thus, there exists corresponding $t \geq 0$ such that $\text{STP}(y, x_2, t)$ is true. Hence, there exists $z \geq \langle x_1, 0 \rangle$, namely $\langle y, t \rangle$, such that $\text{STP}(l(z), x_2, r(z))$ is true.
  Therefore, when $x_2 \in \text{INF}$, for all $x_1$, $\Phi(x_1, x_2, q) \downarrow$, which is $\Phi(x_1, g(x_2)) \downarrow$. Hence, if $x_2 \in \text{INF}$, $g(x_2) \in \text{TOT}$.

- Show $x_2 \in \text{INF} \leftarrow f(x_2) \in \text{TOT}$.
  
  If $x_2 \notin \text{INF}$, there is $y_0$ such that, for all $y \geq y_0$, $\Phi(y, x_2) \uparrow$.
  Therefore, no matter what $y \geq y_0$ and $t \geq 0$ we choose, $\text{STP}(y, x_2, t)$ is false, i.e., there is no $z \geq \langle y_0, 0 \rangle$ such that $\text{STP}(l(z), x_2, r(z))$ is true.
  Therefore, if $x_2 \notin \text{INF}$, for all $x_1 \geq y_0$, $\Phi(x_1, x_2, q) \uparrow$, which is $\Phi(x_1, g(x_2)) \uparrow$. Hence, if $x_2 \notin \text{INF}$, $g(x_2) \notin \text{TOT}$.

Therefore, we conclude $\text{INF} \leq_m \text{TOT}$.

Then, overall, we conclude $\text{INF} \equiv_m \text{TOT}$.

4. Show that there is a number $e$ such that $W_e = \{e\}$. (Q3, p104)

**Ans.**

Consider the following function, which is clearly partially computable.

\[
f(x_1, x_2) = \begin{cases} 
0 & \text{if } x_2 = x_1 \\
\uparrow & \text{otherwise}
\end{cases}
\]

By Recursion Theorem, there is a number $e$ such that

\[
\Phi_e(x_2) = f(e, x_2)
\]

6
Therefore,
\[
\Phi_e(x) = \begin{cases} 
0 & \text{if } x = e \\
\uparrow & \text{otherwise}
\end{cases}
\]

Clearly, \(W_e = \{e\} \).

5. Suppose \(L_1\) and \(L_2\) are regular sets defined over the alphabet set \(A\). Show that the following sets are also regular:

(a) \(\text{Prefix}(L_1) = \{ x \mid \exists y \in A^*, xy \in L_1 \} \).

**Ans.**

Let \(\mathcal{M}_1\) be the deterministic finite state automaton such that \(L(\mathcal{M}_1) = L_1\), where \(\mathcal{M}_1\) is defined by \((Q_1, A, \delta_1, q_1^1, F_1)\).

Prove \(\text{Prefix}(L_1)\) is regular by showing that, given \(\mathcal{M}_1\), we can construct a deterministic finite state automaton \(\mathcal{M}_2\) such that \(L(\mathcal{M}_2) = \text{Prefix}(L_1)\), where \(\mathcal{M}_2\) is defined by \((Q_2, A, \delta_2, q_1^2, F_2)\).

Define \(Q_2 = Q_1\), \(\delta_2 = \text{delta}_1\), and \(q_1^1 = q_1^2\).

Let \(R = \{ z \in A^* \mid |z| < |Q_1| \} \). Then, for each \(q \in Q_1\), let \(q\) be in \(F_2\) if and only if there is \(y \in R\) such that \(\delta_1^*(q,y) \in F_1\). Since \(R\) is a finite set, this procedure is done within finite steps.

Note that, if there is no such \(y \in R\), no final state in \(F_1\) is reachable from the state \(q\), i.e., for all \(x \in A^*\) such that \(\delta_1^*(q_1^1,x) = q\), \(x \not\in \text{Prefix}(L_1)\).

Correctness of the construction is not included.

(b) \(L_1/L_2 = \{ x \mid \exists y \in L_2, xy \in L_1 \} \).

**Ans.**

Let \(\mathcal{M}_1\) and \(\mathcal{M}_2\) be the deterministic finite state automata such that \(L(\mathcal{M}_1) = L_1\) and \(L(\mathcal{M}_2) = L_2\). \(L_1\) and \(L_2\) are defined by \((Q_1, A, \delta_1, q_1^1, F_1)\) and \((Q_2, A, \delta_2, q_1^2, F_2)\), respectively.

Construct a finite state automaton \(\mathcal{M}_3\) such that \(L(\mathcal{M}_3) = L_3\), where \(\mathcal{M}_3\) is defined by \((Q_3, A, \delta_3, q_1^3, F_3)\).

Define \(Q_3 = Q_1 \cup Q_1 \times Q_2\) and \(q_1^3 = q_1^1\).

Define \(\delta_3\) as follows:

- \(\delta_3(q,a) = \delta_1(q,a)\),
  for any \(q \in Q_1\) and \(a \in A\).
• $\delta_3(q, \epsilon) = \langle q, q_1^3 \rangle$, for any $q \in Q_1$.
• $\delta_3(\langle q^1, q^2 \rangle, a) = \langle \delta_1(q^1, a), \delta_2(q^2, a) \rangle$, for any $q^1 \in Q_1$, $q^2 \in Q_2$ and $a \in A$.

Let $R = \{ z \in A^* \mid |z| < |Q_1 \times Q_2| \}$. Then, for each state $\langle q_i^1, q_i^3 \rangle \in Q_3$ for any $i$, let the state be in $F_3$ if and only if there is $y \in R$ such that $\delta_3^*(\langle q_i^1, q_i^3 \rangle, y) = \langle q_j^1, q_k^3 \rangle$ where $q_j^1 \in F_1$ and $q_k^3 \in F_2$.

Correctness of the construction is not included.