SOLUTIONS (3163, 2008) ①

+ MARKING SCHEME

Q1) (a) $x \times 0 = 0$

(2) $x \times (y+1) = x \times y + x$

So only need to show + is prim:

$x + 0 = x$

$x + (y+1) = (x+y)'$.

Also $sg(0) = 0$

(1) $sg(x+1) = 1$.

Then $rm(x,0) = 0$.

$rm(x, y+1) = rm(x,y)' \times sg(|x - rm(x,y)'|)$. (3)

problems sheet

(b) A suitable machine would have program:

$q_0 1 0 q_1$ ← delete leftmost 1

$q_1 0 R q_2$ ← move right in search of further 1's

$[q_2 0 0 q_3]$ ← if not found, print output 0 and halt

$q_2 1 R q_4$ ← if found, prepare to start subroutine

(4) $q_4 1 0 q_5$ } ← subroutine consists of deleting all but one remaining 1's and halting

$q_5 0 R q_4$

new

Given TM computes the function $2(n+1)$ (3)

similar problem sheet

(c) To compute $\psi(x,y)$, first check if $y \leq 1$ (if not $\psi(x,y)\uparrow$). Then check if $y=1$ (if so $\psi(x,y)=0$).
Finally, if $y=0$, input x to U, and wait for an output — in which case compute $\psi(x,y)=0$ again. If U does not halt then $\psi(x,y)$ remains undefined. (3)

Similar problem sheet

Let $f(x) = \mu^{*} y[\psi(x,y)=0]$.
Looking at f, we see that ① f is total ($\psi(x,1)=0$ for all x), and (1)

② $f(x) = \begin{cases} 0 & \text{if } U \text{ halts on input } x \\ 1 & \text{if } U \text{ does not halt on input } x \end{cases}$

So if f is p.c., f is computable, and hence the halting problem for U is solvable — a contradiction. (3)

Q2 (a) A is c.e. iff there is a computable list of all its members: $a_0, a_1, \ldots$ (or $A = \phi$). (1)

Assume A, B c.e., with effective enumerations $a_0, a_1, \ldots$, and $b_0, b_1, \ldots$, respectively. Then (2)

① $a_0, b_0, a_1, b_1, \ldots$ is an effective enumeration of $A \cup B$

② Enumerate $A \cap B$ by enumerating a_s or b_s into $A \cap B$ whenever we find (resp^y.) $a_s = b_i$ or $b_s = a_j$, some $i, j \le s$. (2)

(if A or $B = \phi$, have $A \cup B = A$ or B, $A \cap B = \phi$)

as homework

① Say A, and hence $\overline{A}$, computable. Then $A, \overline{A}$ are c.e. (1)

② Say $A, \overline{A}$ c.e. with effective enumerations $a_0, a_1, \ldots,\ a_0', a_1', \ldots$, respectively.
Effectively decide whether $x \in A$ or not (any $x \in \mathbb{N}$) by enumerating A and $\overline{A}$ until we find an $i(x)$ such that $x = a_{i(x)}$ or $a'_{i(x)}$ ($i(x)$ exists since $\mathbb{N} = A \cup \overline{A}$).
Then $x \in A \iff x = a_{i(x)}$. (2)

as homework

(b) Write

$$y \in \mathrm{Rng}\, \varphi_e \iff \exists \langle s, x \rangle\ \varphi_{e,s}(x) = y \quad (1)$$

$$\langle x, y \rangle \in \mathrm{Graph}\, \varphi_e \iff (\exists s)\ \varphi_{e,s}(x) = y, \quad (1)$$

and then argue that each set is c.e. because it can be written as a Σ^0_1 set.

similar homework

By (ii), if φ_e is p.c. then $\text{Graph}\,\varphi_e$ is c.e. Conversely, say f has c.e. graph. Then to compute $f(x)$, enumerate $\text{Graph}(f)$, and if some $\langle x,y\rangle$ is enumerated, define $f(x)=y$. (2)

new

c) (i) $z \in X \setminus Y \iff \exists s\,(z \in X_s - Y_s) \in \Sigma^0_1$, so c.e.
$X \searrow Y = (X \setminus Y) \cap Y$ = intersection of two c.e. sets, so c.e. (2)

(ii) $x \in X \setminus Y \iff \exists s\,(x \in X_s - Y_s)$ or $x \in (X \setminus Y) \cap Y$
$\iff x \in X - Y$ or $x \in X \searrow Y$
$\iff x \in (X-Y) \cup (X \searrow Y)$ (2)

(iii) Say $X \searrow Y$ finite. Then $(X \setminus Y) - (X - Y)$ is a finite set S, say. Then can enumerate $X-Y$ by enumerating all members of $X \setminus Y$ into $X-Y$, except for those appearing in S. (2)

(iv) $(X \setminus Y) \cap (Y \setminus X) = \emptyset$ since if $x \in X \setminus Y$ and $x \in Y \setminus X$, $x \in X \cap Y$ and x is in X before Y <u>and</u> x is in Y before X (impossible by choice of X_s, Y_s). Also $(X \setminus Y) \cup (Y \setminus X) = X \cup Y$ since $\subseteq$ immediate, and $(X \setminus Y) \cup (Y \setminus X) \supseteq X \cup Y$ since if $x \in X \cup Y$, $x \in X \setminus Y$ or $x \in Y \setminus X$. (2)

similar homework

3] (a) If S simple and computable,

2) have $\bar{S}$ computable, so c.e. $= W_e$ say.

But then W_e infinite $\subseteq \bar{S}$, a contradiction.

<u>Algorithm</u> <u>for enumerating</u> <u>A simple</u>:

(1) For each i, <u>wait</u> for a stage s such that there is a number $x \in W_{i,s}$ with $x > 2i$.

(2) If such an x appears, enumerate <u>just one</u> such x into A.

<u>Then check</u>: (a) **A c.e.** — immediate from algorithm

(b) **If W_i infinite, $W_i \cap A \neq \phi$** (each i)

<u>PROOF</u>:

Say W_i infinite. Then eventually get some $x \in W_{i,s}$ with $x > 2i$. So $x \in W_i \cap A$

(c) **For each i, A has at most i members $\leq 2i$ — so $\bar{A}$ is infinite.**

<u>PROOF</u>: Since we can only enumerate a number x into A with $x \leq 2i$ on <u>behalf</u> of some W_j with $j < i$, result follows. □

10) ... A simple. □

(b) If $\varphi_{\varphi_x(x,\vec{y})} = \begin{cases} \varphi_{\varphi_x(x,\vec{y})} & \text{if } \varphi_x(x,\vec{y})\downarrow \\ \text{totally undefined} & \text{o.w.} \end{cases}$

can find a computable d s.t. $\varphi_{d(x,\vec{y})} = \varphi_{\varphi_x(x,\vec{y})}$

Then $f(d(x,\vec{y}),\vec{y})$ is computable, $= \varphi_e(x,\vec{y})$, some e.

So $\varphi_{f(d(x,\vec{y}),\vec{y})} = \varphi_{\varphi_e(x,\vec{y})}$.

Taking $x = e$, get

$$\varphi_{f(d(e,\vec{y}),\vec{y})} = \varphi_{\varphi_e(e,\vec{y})} = \varphi_{d(e,\vec{y})}.$$

So have $\varphi_{k(\vec{y})} = \varphi_{f(k(\vec{y}),\vec{y})}$ with $k(\vec{y}) = d(e,\vec{y})$.

Hence $W_{k(\vec{y})} = \text{dom } \varphi_{k(\vec{y})} = \text{dom } \varphi_{f(k(\vec{y}),\vec{y})}$

$$= W_{f(k(\vec{y}),\vec{y})}.$$

(8)

in problem sheet

24) (a) $A \leq_m B$ iff $f(A) \subseteq B$, $f(\bar{A}) \subseteq \bar{B}$, some computable f. (1)

(i) (1) $A \leq_m A \oplus B$ via $f(x) = 2x$, and
$B \leq_m A \oplus B$ via $g(x) = 2x+1$. (2)

(2) Say $A \leq_m C$ via f computable,
$B \leq_m C$ via g "

Then $A \oplus B \leq_m C$ via h where

$$h(x) = \begin{cases} f\left(\frac{x}{2}\right) & \text{if } x \text{ even,} \\ g\left(\frac{x-1}{2}\right) & \text{if } x \text{ odd.} \end{cases}$$ (3)

(ii) By (i) (1), $\underset{\sim}{a}_m \leq \underset{\sim}{a}_m \cup \underset{\sim}{b}_m$, $\underset{\sim}{b}_m \leq \underset{\sim}{a}_m \cup \underset{\sim}{b}_m$,
so $\text{lub}\{\underset{\sim}{a}_m, \underset{\sim}{b}_m\} \leq \underset{\sim}{a}_m \cup \underset{\sim}{b}_m$.

By (i) (2) - If $\underset{\sim}{a}_m \leq \underset{\sim}{c}_m$, $\underset{\sim}{b}_m \leq \underset{\sim}{c}_m$, then $\underset{\sim}{a}_m \cup \underset{\sim}{b}_m \leq \underset{\sim}{c}_m$.
So $\underset{\sim}{a}_m \cup \underset{\sim}{b}_m \leq \text{lub}\{\underset{\sim}{a}_m, \underset{\sim}{b}_m\}$ — so it follows that $\underset{\sim}{a}_m \cup \underset{\sim}{b}_m = \text{lub}\{\underset{\sim}{a}_m, \underset{\sim}{b}_m\}$. (3)

bookwork + similar problem sheet

(b) $A \leq_T B$ iff there is an algorithm, which given $x \in \mathbb{N}$, computes whether or not $x \in A$ via a computation using at most finitely many answers to questions of the form "is $y \in B$?"

(1) $A \equiv_T B \iff A \leq_T B$ & $B \leq_T A$.

notes

(i) If X is A-computable, either $X = \phi$ (so A-c.e.)

or can define an A-computable f by

(3) $f(0) = \text{least } x \in X$

$f(n+1) = \text{least } x \in X \text{ with } x > f(n)$,

s.t. $X = \text{range } f$ (so A-c.e.)

(ii) ($\Rightarrow$) X A-computable $\Rightarrow$ $\overline{X}$ A-computable.

So $X, \overline{X}$ A-c.e. (by (i))

($\Leftarrow$) If X or $\overline{X} = \phi$, result follows.

o.w. let $X = \text{range } f$, f A-computable

$\overline{X} = \text{range } g$, g A-computable

(3) Then $x \in X \Leftrightarrow f(\mu y[f(y) = x \vee g(y) = x]) = x$.

(iii) ① X is A-c.e. $\Rightarrow$ $X = W_e^A$, some e

$\Rightarrow X = \{x \mid x \in W_e^A\} \Rightarrow X = \{x \mid \exists s\, [\underbrace{x \in W_{e,s}^A}_{A\text{-computable}}]\}$

$\Rightarrow X \in \Sigma_1^A$.

② Conversely, $X \in \Sigma_1^A \Rightarrow X = \{x \mid \exists s\, R^A(x,s)\}$ some A-computable R^A,

Define $\psi^A(x) = \begin{cases} 0 & \text{if } \exists s\, R^A(x,s) \\ \text{undefined} & \text{o.w.} \end{cases}$

Then ψ is a partial function computable from A.

(4) So $\psi^A = \Phi_e^A$, some e, giving $X = \text{dom } \psi^A$ $= \text{dom } \Phi_e^A = W_e^A$ — so X is A-c.e.

all in previous spaces

Q 5) (a) Let (A, B, Φ) be a typical member of a list of all pairs $A \neq B$ taken from our infinite list of Turing incomparable sets, and p.c. functionals Φ.

We satisfy all requirements of the form $R = R_{A,B,\Phi}$: $A \neq \Phi^B$.

Build sequences $\sigma_0 \subset \sigma_1 \subset \ldots\ldots \subset A$
$\tau_0 \subset \tau_1 \subset \ldots\ldots \subset B$.

The construction Let (A, B, Φ) be the i^{th} such triple.

Stage 0 . Define $\sigma_0 = \tau_0 = \phi$ (empty string)

Stage $s+1 = i+1$. (assuming σ_s, τ_s already defined)

Let $x = |\sigma_s|$. Look for $\tau \supset \tau_s$ s.t. $\Phi^\tau(x)\downarrow$.

Case I. τ exists, with $\Phi^\tau(x)\downarrow = k$, say.

Define $\sigma_{s+1} = \sigma_s ^\frown (1-k)$ $\tau_{s+1} = \tau \supset \tau_s$

Case II. No such τ exists

Define $\sigma_{s+1} = \sigma_s ^\frown 0$, $\tau_{s+1} = \tau_s ^\frown 0$.

(Stage $s+1 = 2i+1$ is similar)

Finally define $A = \bigcup_s \sigma_s$, $B = \bigcup_s \tau_s$.

Similar problem sheet

Then – ① $R_{A,B,\Phi}$ is satisfied for all A, B, Φ

Pf: Say $A = \Phi^B$.

Then at stage $s+1 = i+2$ of the construction, have $|\sigma_s| = x$, and $A(x) = \Phi^B(x)$ — so $\exists\, \tau \supset \tau_s$ s.t. $\tau \subset B$ and $\Phi^\tau(x)\downarrow$.

But in that case, by Case I, we have $\tau_{s+1} = \tau$, with $\Phi^\tau(x)\downarrow = k$, say,

so that $\Phi^B(x)\downarrow = \Phi^{\tau_{s+1}}(x) = k$, but

$$A(x) = \sigma_{s+1}(x) = 1 - k \neq \Phi^B(x) \quad \boxtimes$$ (15)

Skip — proven already

To see $\underset{\sim}{a}, \underset{\sim}{b} = deg(A), deg(B)$ resp'y $\leq \underset{\sim}{0}'$, just need to check the construction can be carried out using an oracle for K_0. But construction is effective, apart from search for τ.

But τ exists at stage $s+1$

$\Leftrightarrow (\exists\, t, \tau)[\tau \supset \tau_s \;\&\; \Phi^\tau_t(|\sigma_s|)\downarrow] \in \Sigma^0_1$ (approx at stage t)

— so can be carried out using an oracle for K_0 $\square$

(5)

18

Q6.

Some topics, a selection of which might be touched on in such an essay:

(1) The importance of algorithms in the history of mathematics, and the way science focuses on identifying algorithmic content in the material universe.

(2) The programmes of Leibniz, Hilbert for algorithmically capturing logic and mathematics.

(3) The various formulations of the notion of a computable function, and their importance to an examination of the feasibility of Hilbert's programme.

(4) The notion of a Turing machine and how a TM computes, and how to obtain a Universal Turing Machine U.

(5) How such a U can be used to reveal the existence of incomputable sets and unsolvable problems. Links with logic.

(6) Computably enumerable sets. Incomputability and the arithmetical hierarchy (the connection with Σ_1^0 sets).

(7) How U anticipates the stored program computer.

(8) The Church-Turing Thesis ...

(9) ... and its extensions to the real world — evidence either way?

(10) The human mind as a Turing machine?

(11) Hilbert's 10th Problem, and the search for diverse natural examples of incomputable sets.

(12) Oracle TMs, and the resulting structure (the Turing universe).

(13) The Turing universe as a model for computably complex environments.

Have been warned of this question

20