1. $[\mathbf{2+2+3+3+2+3+3+2+2+3=25}]$

Answer with a short justification whether the following statements are *true* or *false*. No credit will be given for writing only *true* or *false*.

(a) A set may be equinumerous to its proper subset.

Ans. The statement is true. If the set is infinite then a proper subset may be equinumerous to the set. As an example - $\mathbb{N} \equiv \mathbb{E}$**. The function** $f : \mathbb{N} \longrightarrow \mathbb{E}$, $f(n) = 2n$**, is a bijection.**

(b) No *left-inverse* of a map $f : A \longrightarrow B$ is present if it is not *one-to-one*.

Ans. Let $f : A \longrightarrow B$ **is not one-to-one, but** $g : B \longrightarrow A$ **is a left-inverse of** f **i.e. there exists** a_0 **and** a_1 **such that** $f(a_0) = f(a_1)$ **and also** $g \circ f = 1_A$**. But then** $g(f(a_0)) = g(f(a_1))$**, implies** $a_0 = a_1$**, a contradiction. Hence the statement is true.**

(c) Given the map $f : A \longrightarrow B$, we define a map $F : \mathcal{P}B \longrightarrow \mathcal{P}A$, so that $F(D) = \{c \in A \ : \ f(c) \in D\}$, $D \subseteq B$. The map F is monoton i.e. $C \subseteq D \Rightarrow F(C) \subseteq F(D)$.

Ans. $F(C) = \{c \in A : f(c) \in C\} = \{c \in A : f(c) \in C \subseteq D\} \subseteq \{c \in A : f(c) \in D\} = F(D)$**. Hence the statement is true.**

(d) The collection of all finite subsets of natural numbers, $\mathbf{2}^{\mathbb{N}}_{fin} = \{A : \ A \subset \mathbb{N} \text{ and } A \text{ is finite}\}$, is not denumerable.

Ans. The statement is false. Let $f : \mathbf{2}^{\mathbb{N}}_{fin} \longrightarrow \mathbb{N}$ **be defined as follows.**

$$f(A) = \left\{ \begin{array}{ll} 0 & \textbf{if } A = \emptyset, \\ 2^{a_1} + \cdots + 2^{a_k} & \textbf{if } A = \{a_1, \cdots, a_k\} \ . \end{array} \right\}$$

It can be shown to be a bijection.

(e) A denumerable set of variables can be defined inductively using a *finite alphabet*.

Ans. Consider the alphabet $\{x, 0\}$**. Variable names are defined inductively as follows.**

- **Basis:** x **is a variable name.**
- **Induction: If** v **is a variable name then so is** $v0$**.**
- **Nothing else is a variable name.**

The variable names are $\{x, x0, x00, x000, \cdots\}$**.**

(f) Fifteen (15) different equivalence relations can be defined on a set of four elements $A = \{a, b, c, d\}$. [Note that an equivalence relation divides the set into nonempty parts.]

Ans. The statement is true. The different partitions are.

$\{\{a\},\{b\},\{c\},\{d\}\}$	$\{\{a,b\},\{c,d\}\}$	$\{\{a,c\},\{b,d\}\}$	$\{\{a,d\},\{b,c\}\}$
$\{\{a\},\{b,c,d\}\}$	$\{\{b\},\{a,c,d\}\}$	$\{\{c\},\{a,b,d\}\}$	$\{\{d\},\{a,b,c\}\}$
$\{\{a\},\{b\},\{c,d\}\}$	$\{\{a\},\{c\},\{b,d\}\}$	$\{\{a\},\{d\},\{b,c\}\}$	$\{\{b\},\{c\},\{a,d\}\}$
$\{\{b\},\{d\},\{a,c\}\}$	$\{\{c\},\{d\},\{a,b\}\}$	$\{\{a,b,c,d\}\}$	

(g) The λ-term $(\lambda xyz.xz(yz))(\lambda ab.a)(\lambda p.pp)(\lambda xy.y)$ can be reduced to '**false**'.

Ans. The statement is true.

$$\begin{aligned}(\lambda xyz.xz(yz))(\lambda ab.a)(\lambda p.pp)(\lambda xy.y) &\to_\beta (\lambda yz.(\lambda ab.a)z(yz))(\lambda p.pp)(\lambda xy.y)\\ &\to_\beta (\lambda yz.(\lambda b.z)(yz))(\lambda p.pp)(\lambda xy.y)\\ &\to_\beta (\lambda yz.z)(\lambda p.pp)(\lambda xy.y)\\ &\to_\beta (\lambda z.z)(\lambda xy.y)\\ &\to_\beta (\lambda xy.y)\\ &= \textsf{false}\end{aligned}$$

(h) Reflexive-transitive closure of a binary relation R on a set A is always an equivalence relation.

Ans. The statement is false because the relation may not be symmetric.

(i) Every β-reduction of any λ-term will give a β-normal form.

Ans. The statement is false. The lambda term $(\lambda x.xx)(\lambda x.xx)$ does not have any β-normal form.

(j) The decision problem, "the vertex **d** is reachable from the vertex **s** in a undirected graph G", can be translated to a decision problem of a language.

Ans. A graph G is represented as a 2-tuple of data (V, E). The collection of encodings of (G, s, d) over an alphabet $\{0, 1, *\}$, so that d is reachable from s in G is a language over $\{0, 1, *\}$. Therefore the decision problem of graph is a decision problem of a language.

2. **[5]**

Let the collection of C* programs that calculate functions on the set of natural numbers, $\mathbb{N} \to \mathbb{N}$, be $\mathcal{C}_\mathbb{N}$. Prove by direct use of Cantor's diagonal argument that $\mathcal{C}_\mathbb{N}$ cannot be *equinumerous* to the collection of all functions on natural numbers ($\mathbb{N}^\mathbb{N}$).

Ans. Let each function from $\mathbb{N}$ to itself be computed by a C* program. Let the program P_i computes the function f_i, $i \in \mathbb{N}$. This can be so written as the collection of C* programs and the set of natural numbers are equinumerous. Consider the function $f : \mathbb{N} \longrightarrow \mathbb{N}$ defined as follows.

$$f(n) = \begin{cases} 5 & \textbf{if } f_n(n) \neq 5 \\ 6 & \textbf{if } f_n(n) = 5. \end{cases}$$

The function f cannot be identified with any f_i

$$f(n) = 5 \textbf{ iff } f_n(n) \neq 5, \textbf{ for all } n \in \mathbb{N}.$$

Therefore all functions on $\mathbb{N}$ cannot be computed by a C* program.

3. **[2 + 2 + 1]**

Give an example of a *fixed point combinator* other than the Curry and the Turing combinators. Show that it is a *fixed point combinator*. Justify that there are infinite many of them.

Ans. Let $A = \lambda cury.y(curry)$ **and** $F = AAAA$. **We get**

$$Fu = AAAAu \; = \; (\lambda cury.y(curry))AAAu \; \to_\beta \; u(AAAAu) \; = \; u(Fu).$$

Let $v_1, v_2, \cdots, v_k$ **be variables such that only two of them (except the** k**th one) are same i.e.** $v_i = v_j, 1 \leq i < j < k$. **We define** $A = \lambda v_1 \cdots v_i \cdots v_{j-1}v_{j+1} \cdots v_k.v_k(v_1 \cdots v_k)$ **and** $F = \overbrace{A \cdots A}^{k-1}$. F **is a fixed point combinator.**

4. **[10]**

Give an inductive definition of '**remainder m n**' ($m\%n$). Justify that *remainder* is λ-definable. Start from the *Barendregt numeral* and define everything that is required to define *remainder*.

Ans. Barendregt Numerals: $0 \equiv \lambda x.x$ **and** $n + 1 \equiv (F, n) = \lambda x.xFn$.

(a)

$$\mathbf{rem}\ m\ n \; = \begin{cases} m & \text{if (less } m\ n), \\ \mathbf{rem}\ (\mathbf{min}\ m\ n)n & \text{else.} \end{cases}$$

Fixed point of : $\lambda f.\lambda mn.(\text{less } m\ n)\ m\ (f\ (\mathbf{min}\ m\ n)\ n)$.

(b)

$$\text{less } m\ n \; = \begin{cases} T & \text{if (and (isZero } m\text{) (not (isZero } n\text{))),} \\ F & \text{if (isZero } n\text{),} \\ \text{less (pred } m\text{) (pred } n\text{)} & \text{else.} \end{cases}$$

The lambda term is easy.

(c)

$$\mathbf{min}\ m\ n \; = \begin{cases} 0 & \text{if isZero } m, \\ m & \text{if (isZero } n\text{),} \\ \text{min (pred } m\text{) (pred } n\text{)} & \text{else.} \end{cases}$$

Again a fixed point.

(d) Definitions of and, pred, isZero are already known.