Mathematical Logic Assignment 1

522030910158 Song Yuanyi

October 2024

1.(a) From $f : \mathbb{N} \to A$, we can make a listing $f(0), f(1), \ldots, f(n), \ldots$. Because f is surjective, this listing include all members of A and maybe have repetition. Construct $g(a_i)$ = the position of the first occurrence in the listing for $\forall a_i \in A$. If A is finite, it must be countable. If A is infinite, we only need to list $g(a_i)$ in order, which is enumerable so countable.

1.(b) From $f : \mathbb{N} \to A$, we have $\forall a_i \in A, \exists n \in \mathbb{N}, f(a_i) = n$. Namely, every a_i can map only one nature number but different a_i can map to the same nature number. From f is surjective, we have $\forall n \in \mathbb{N}, \exists a_k \in A, f(a_k) = n$, namely every nature number be mapped to a elemnet of A. Therefore, $\mathbb{N} \preceq A$. $\mathbb{N}$ is infinite so A is infinite.

2. Let ${}^{\mathbb{N}}\mathbb{N} = \{f_i | i = 0, 1, 2, \ldots\}$ is the set of function mapping from $\mathbb{N}$ to $\mathbb{N}$. Obviously, $\forall f_i \in^{\mathbb{N}} \mathbb{N}, f_i : \mathbb{N} \to S \subset \mathbb{N}$. For every $n \in \mathbb{N}$, $f_i(n)$ is correspondent to some nature number which is in S. Construct function f_x s.t. $f_x(i) \neq f_i(i), i \in \mathbb{N}$. We can list ${}^{\mathbb{N}}\mathbb{N}$ as a table whose row are functions f_i and columns are nature numbers 0,1,2,.... Therefore by diagonal argument, f_x is not in the table. If $f_x \in^{\mathbb{N}} \mathbb{N}$, it is contradicted with $f_x(x) \neq f_x(x)$. If $f_x \notin^{\mathbb{N}} \mathbb{N}$, f_x satisfies the demand of function, which is contradicted. To sum up, ${}^{\mathbb{N}}\mathbb{N}$ is uncountable.

3. Given a constant number, it is easy to determine if it is a perfect number. First we can find its divisors, then add them to see that if the summation equals to the number itself. For example, 28=1x28=2x14=4x7=1+2+4+7+14, 35=1x35=5x7$\neq$ 1+5+7.

1. give input i
2. find divisors of i : for k = 1 : i / 2
   (a) if i/k is integer, k and i/k is divisors
   (b) k++
3. sum the divisors and avoid repetitions

4. compare summation and i
    (a) if sum==i, output yes
    (b) else, output no

The existence of this algorithm shows the set of perfect numbers is effectively decidable.

4. $A \subset \mathbb{N}$. Give a number x, first determine if x is a nature number. Subsequently, compare x with one member of A and one member of $\mathbb{N} \setminus A$ in every loop in the order of listing (from effectively enumerable) to determine whether or not $x \in A$. It is obvious that $x \in A$ or $x \in \mathbb{N} \setminus A$, so this algorithm can halt. Assume the the algorithms of listing the member of A and $\mathbb{N} \setminus A$ are $\mathcal{A}(\cdot)$ and $\mathcal{B}(\cdot)$ respectively. '$\cdot$' means the position of listing (ordered from 0).

1. give input x
2. i = 0
3. if $x = \mathcal{A}(i)$, decide $x \in A$ (output yes) , the algorithm halts
4. if $x = \mathcal{B}(i)$, decide $x \notin A$ (output no) , the algorithm halts
5. i++
6. back to 3

5. First analyse the relationship between P and R. $P = \{n \in \mathbb{N} | \forall x < n, x \in R\}$, set P contains the elements which all of the nature number smaller than is in R . So P means the continuous members from 0 to $x+1$ in R , for example if $R = \{0, 1, 2, 3, 5, 6, 8\}$ where $x = 3$ then $P = \{0, 1, 2, 3, 4\}$. We can construct such algorithm, maintain a variable x (same as example) to mark the end number of continuous sequence with the beginning number 0 in R . As long as s is updated, we can print newly discovered members of P. The algorithm should halt when the algorithm of listing R halt and assume this algorithm is $\mathcal{A}$. Therefore, P is effectively enumerable.

1. let $x = 0$ to mark, $i = 0$ to count(list), a bool array arr=[false] and arr[k] record the existence of k in R
2. loop i with stride 1 (i=i+1)
    (a) arr[$\mathcal{A}(i)$] = true
    (b) if $\mathcal{A}(i) = x+1$, update x : check elements of arr from arr[x+1], $x = \max\{x | \forall i \leq x, arr[i] = true\}$ and print from old $x+2$ to new $x+1$
3. halt when $\mathcal{A}$ halt