Lista 3 de Problemas y Ejercicios
Lógica Matemática

Cristo Daniel Alvarado

22 de noviembre de 2024

3.1. Ejercicios

Ejercicio 3.1.1
Demuestre que todo subconjunto cofinito de $\mathbb{N}$ (es decir, cuyo complemento es finito) es computable.

Demostración:
Sea $X \subseteq \mathbb{N}$ un conjunto cofinito, entonces su complemento $\mathbb{N} \setminus X$ es finito. Sea $N \in \mathbb{N}$, se tiene que la función característica $\chi_{\{N\}}$ es computable, pues tiene como algoritmo:

```
int chi_N(int n){
    if(n == N) return 1;
    else return 0;
}
```

por lo que el conjunto $\{N\}$ es computable, en particular el conjunto:

$$\mathbb{N} \setminus X = \left\{ n \in \mathbb{N} \middle| n \notin X \right\}$$

es computable (por ser finito) ya que es unión finita de conjuntos numerables, luego su complemento el cual es X es computable. $\blacksquare$

Ejercicio 3.1.2
Suponga que $X \subseteq \mathbb{N}$ es computable, y sea $f : \mathbb{N} \to \mathbb{N}$ una función total computable. Demuestre que:

$$f^{-1}[X] = \left\{ n \in \mathbb{N} \middle| f(n) \in X \right\}$$

es un conjunto computable.

Demostración:
Considere el siguiente algoritmo de la función característica de $f^{-1}[X]$:

```
int f_1_[X](int n){
    if(chi_X(f(n))) return 1;
    else return 0;
}
```

como f es total computable, entonces $f(n)$ existe para todo n, luego al ser X un conjunto computable, en una cantidad finita de tiempo se obtiene si χ_X evaluada en $f(n)$ es cero o uno, en cuyo caso se retorna cero o uno en el algoritmo definido anteriormente, el cual siempre retorna algo. $\blacksquare$

Ejercicio 3.1.3
Defina la función $f : \mathbb{N} \to \mathbb{N}$ mediante:

$$f(n) = \begin{cases} 1 & \text{si la expansión decimal de } \pi \text{ contiene una sucesion de al menos } n \text{ digitos} \\ & \text{consecutivos iguales a 7.} \\ 0 & \text{en otro caso.} \end{cases}$$

Demuestre (sin usar ningún hecho especial sobre π) que la función f es total computable.

Demostración:
Es inmediato del siguiente algoritmo:

```
int f(int n){
    recorrer digito por digito la expansion decimal de pi hasta encontrar un 7{
        int cont = 1;
        cont cuenta el numero de 7s despues del primer 7;
        if(n <= cont) return 1;
    }
}
```

que f es computable. Si f no fuese total computable (es decir, que f no sea la función constante uno), entonces existiría al menos un $N \in \mathbb{N}$ tal que $f(N)$ no está bien definido, por la forma en que definimos el algoritmo de f, se tendría que $N+1, \ldots$ tampoco estarían bien definidos. Sea n_0 el mínimo entero no negativo tal que $f(n_0)$ no está bien definido (es decir que el algoritmo anterior sigue funcionando). Construímos el algoritmo:

```
int f_2(int n){
    if(n < n_0) return f(n);
    else return 0;
}
```

esta es el algoritmo de la función f, mismo que es total. ■

Observación 3.1.1
¿Puedo elegir tal n_0 en la demostración anterior?

Ejercicio 3.1.4
Suponga que $g : \mathbb{N} \to \mathbb{N}$ es una función no-creciente, es decir que $g(n+1) \leq g(n)$ para todo $n \in \mathbb{N}$. Pruebe que g debe ser total computable.

Demostración:
Primero veamos que g es computable. ■

Ejercicio 3.1.5
Demuestre que la función $f : \mathbb{N}^3 \to \mathbb{N}$ dada por:

$$f(x,y,z) = \begin{cases} y & \text{si} \quad x = 0. \\ z & \text{si} \quad x \neq 0. \end{cases}$$

es total computable.

Demostración:
■

Ejercicio 3.1.6
Considere una retícula de calles que conste de n calles que van de este a oeste, atravesadas por m calles que van de norte a sur, de tal suerte que se genere un mapa rectangular con mn intersecciones. Si un peatón se propone caminar (utilizando dichas calles) para llegar desde la esquina noreste hasta la suroeste, caminando únicamente hacia el este o hacia el sur, y cambiando de dirección únicamente en las esquinas, denote por $r(n,m)$ a la cantidad de posibles rutas que nuestro peatón puede tomar. Demuestre que la función $r : \mathbb{N}^2 \to \mathbb{N}$ es total computable.

Demostración:

$\blacksquare$

Ejercicio 3.1.7
Proporcione un ejemplo de una función no-total, $g : \mathbb{N}^2 \to \mathbb{N}$, tal que la función $h : \mathbb{N} \to \mathbb{N}$ obtenida por medio de una búsqueda no acotada:

$$h(x) = (\mu y)(g(x, y) = 0)$$

sí es total.

Solución:

$\square$

Definición 3.1.1
El operador μ significa **el mínimo tal que**, en caso de que exista (y la función queda sin definir en caso de que no). El acto de invocar a μ se conoce como búsqueda no acotada.

Ejercicio 3.1.8
Demuestre que si $X \subseteq \mathbb{N}$ es el conjunto de números de Gödel de máquinas de Turing (es decir, $n \in X$ si y sólo si $\varphi(n, \cdot)$ es una máquina de Turing válida, en donde φ es la máquina de Turing universal), entonces la función característica χ_X es total computable.

Demostración:

$\blacksquare$