Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
IIC2213 – Lógica para Ciencias de la Computación – 2022-1
Ayudante: Diego Bustamante

Ayudantía 2 - Decidibilidad y Reducciones

Pregunta 1

Argumente si los siguientes lenguajes/problemas son decidibles o indecidibles.

i) $L = \mathbb{N}$ pares - Decidible.

ii) $L = \mathbb{Q}$ - Decidible, se pueden codificar como fracciones.

iii) $L = \{\pi\}$ - Decidible, se puede codificar en el alfabeto.

iv) $L = \mathbb{I}$ - Indecidible, no se pueden codificar un número infinito de símbolos especiales ni aceptar inputs infinitos. El problema de los irracionales es que siempre será un problema codificarlos.

v) $L =$ números primos - Decidible, se puede aceptar un primo de 10^{100} dígitos y eventualmente el algoritmo terminará.

vi) $L =$ cualquier conjunto finito - Decidible, se puede "hardcodear".

vii) $L = 2^{\mathbb{N}}$, todos los subconjuntos de los naturales - Indecidible, no se aceptan inputs infinitos.

viii) Ejecutar cualquier MT sobre cualquier Input de su Alfabeto - Indecidible, como hay máquinas que no terminan, nuestra MT universal no terminaría al ejecutarlas.

ix) Resolver el Ajedrez - Decidible, el número de juegos es finito debido a las reglas del Ajedrez. Shannon lo estima en 10^{120} posibles juegos, pero eventualmente se pueden computar todos.

x) Encontrar una solución a toda EDO - Indecidible, queda propuesto investigar sobre las reducciones que existen. Es un problema parecido al de los irracionales ¿cómo codificar todas esas funciones especiales?

Más problemas indecidibles (link): Mortality, Corresponcencia, si una CFG es ambigua, etc.

Pregunta 2

Argumente si las siguientes funciones/algotirmos son computables.

i) $f(w_d) = w_b$, función que convierte una codificación decimal a binario - Computable, se puede programar el algoritmo.

ii) $f(w) = \sqrt{w}$ - No computable, para algunos inputs no termina.

iii) $f(w) = w^{inv}$ (invertir el input) - Computable, se puede programar con un poco de dedicación en una MT.

iv) $f(w) = |w|$ - Computable se puede generar un contador e ir sumándole 1 por cada carácter borrado.

v) $f(w) = \pi_w$, función que convierte un entero en el dígito número w de π - Computable, existen algoritmos para calcular π con la precisión que uno quiera y despues se puede buscar el dígito coorespondiente (no entra en un loop infinito de cálculos).

vi) $f(w) = w_1\#w_2\#...\#w_n$, donde cada w_i es un posible slice de w - Computable, todo es finito así que se puede enumerar todo slicing de input.

Pregunta 3

Reducir L_2 a L_1.

$L_1 = \{w \mid \exists M1, M2.w = C(M1)0000C(M2), M1\ acepta\ C(M2) \vee M2\ acepta\ C(M1)\}$

$L_2 = \{w \mid \exists M.w = C(M), M\ acepta\ C(M)\}$

$f(w) = w0000w$, esto nos permite argumentar que algunos casos indecidibles de L_2 están en L_1, por lo que este lenguaje también es indecidible.

Nota: faltan algunas discusiones que se dieron durante la ayudantía. ¡Recomiendo fuertemente pedir apuntes, conversar sobre los problemas con los demás estudiantes y asistir al resto de las ayudantías!