Jerarquía de la Computabilidad (segunda parte)

Comentario: Ninguno reviste mucha dificultad. Hacer mínimamente los ejercicios 1 al 4.

Ejercicio 1. Probar que los lenguajes L_U = {(<M>, w) | M acepta w}, y HP = {(<M>, w) | M para sobre w} pertenecen a la clase RE.
Ayuda: construir MT.

Ejercicio 2. Responder cada uno de los incisos.

a. ¿Se puede decidir si una MT M con una cinta, a partir de la cadena vacía λ, escribe alguna vez un símbolo no blanco?
Ayuda: ¿Cuántos pasos puede hacer M antes de entrar en loop?

b. ¿Se puede decidir si a partir de un input w, una MT M que sólo se mueve a la derecha para?
Ayuda: ¿Cuántos pasos puede hacer M antes de entrar en loop?

c. ¿Se puede decidir si dada una MT M, existe un input w a partir del cual M para en a lo sumo 10 pasos?
Ayuda: ¿Hasta qué tamaño de cadenas hay que chequear?

d. ¿Se puede decidir si dada una MT M, existe un input w de a lo sumo 10 símbolos a partir del cual M para?
Ayuda: ¿En este caso se puede acotar la ejecución de M considerando la cantidad de pasos, la cantidad de celdas recorridas u otro parámetro?

Ejercicio 3. Explicar cómo enumeraría los números naturales pares, los números enteros, los números racionales (o fraccionarios) y las cadenas de Σ* siendo Σ = {0, 1}.

Ejercicio 4. Una función f : A ⟶ B se dice que es total computable, si y sólo si existe una MT M_f que computa f para todo elemento a ∈ A. Sea la función f_{HP} : Σ* ⟶ {0, 1}, tal que:

f_{HP} (x) = 1, si x = (<M>, w) y M para a partir de w.

f_{HP} (x) = 0, si x = (<M>, w) y M no para a partir de w, o bien x ≠ (<M>, w).

Probar que la función f_{HP} no es total computable.
Ayuda: Se podría probar que asumiendo que f_{HP} es total computable, se llega a que HP es recursivo.

Ejercicio 5. Responder cada uno de los incisos.

a. Si $L_1 \in$ RE y $L_2 \in$ RE, ¿$L_1 - L_2 \in$ RE?
b. Si $L_1 \cap L_2 \in$ RE, ¿L_1 o $L_2 \in$ RE?
c. Si $L_1 \cup L_2 \in$ RE, ¿L_1 o $L_2 \in$ RE?

Ejercicio 6. Explicar (informal pero claramente) cómo sería una MT que genera la n-ésima fórmula booleana satisfactible, cuya sintaxis contiene variables de la forma x_i, los operadores lógicos del conjunto {¬, ∧, ∨}, y paréntesis.