Ejercicios relacionados con las clases teóricas y prácticas 3 y 4

A. EJERCICIOS PARA ENTREGAR (ninguno reviste mayor dificultad):

Ejercicio 1. Probar que el lenguaje L_U = {(<M>,w) | M acepta w} pertenece a la clase RE. *Ayuda: la prueba es similar a la desarrollada en la clase 3 para demostrar que D = { w_i | M_i acepta w_i } $\in$ RE.*

Ejercicio 2. Explicar informal pero claramente cómo trabajaría una MT que genera la n-ésima fórmula booleana satisfactible (es decir que existe una asignación de valores de verdad que la hace verdadera), cuya sintaxis contiene variables de la forma x_i, los operadores lógicos del conjunto {¬, ∧, ∨} y paréntesis.

Ejercicio 3. Justificar informal pero claramente cada uno de los incisos siguientes:
a. Se puede decidir si una MT M, 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, dada una MT M, si 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?*

Ejercicio 4. Considerando la reducción de HP a L_U descripta en clase, responder:
a. Explicar por qué la función identidad, es decir la función que a toda cadena le asigna la misma cadena, no es una reducción de HP a L_U.
b. Explicar por qué las MT M´ generadas en los pares (<M´>,w) de L_U, o bien paran aceptando, o bien loopean.
c. Explicar por qué la función utilizada para reducir HP a L_U también sirve para reducir HP^C a L_U^C.
d. Explicar por qué la función utilizada para reducir HP a L_U no sirve para reducir L_U a HP.
e. Explicar por qué si el input v de la MT M_f que computa la función de reducción no tiene la forma (<M>,w), no es correcto que M_f genere, en lugar de la cadena 1, un par de la forma ($<M_{\Sigma^*}>$, v), siendo M_{Σ^*} una MT que acepta todas las cadenas.
f. Explicar por qué la siguiente MT M_f no computa una reducción de HP a L_U: dado v,
   - Si v no tiene la forma (<M>,w), entonces M_f genera el output 1.
   - Si v tiene la forma (<M>,w), entonces M_f ejecuta M sobre w, y: si M acepta w entonces genera el output (<M>,w), y si M rechaza w entonces genera el output 1.

Ejercicio 5. Considerando la reducción de L_U a L_{Σ^*} descripta en clase, responder:
a. Explicar por qué no sirve como función de reducción la función siguiente: a todo input le asigna como output el código $<M_{\Sigma^*}>$.
b. Explicar por qué la reducción descripta en clase no sirve para probar que $L_{\Sigma^*} \notin$ RE.

Ejercicio 6. Probar formalmente que las funciones de reducción gozan de la propiedad transitiva. *Ayuda: revisar la idea general comentada en clase.*

Ejercicio 7. Sea el lenguaje D_{HP} = {w_i | M_i para desde w_i, según el orden canónico}. Encontrar una reducción de D_{HP} a HP.

Ejercicio 8. Sean VAL y UNSAT los lenguajes de las fórmulas booleanas válidas e insatisfactibles (todas y ninguna asignación de valores de verdad las hace verdaderas, respectivamente). Encontrar una reducción de VAL a UNSAT.

B. EJERCICIOS QUE NO SON PARA ENTREGAR (se sugiere analizarlos):

Ejercicio. Recordar cómo probamos en la clase 2 que asumiendo R $\subset$ RE se cumple RE $\subset \mathfrak{L}$.

Ejercicio. Probar que el lenguaje HP = {(<M>,w) | M para sobre w} pertenece a la clase RE. *Ayuda: la prueba es similar a la desarrollada en la clase 3 para demostrar que D = { w_i | M_i acepta w_i } $\in$ RE.*

Ejercicio. Construir una MT que genere todos los índices i tales que (<M_i>, w_i) $\in$ HP, según el orden lexical canónico.

Ejercicio. En la clase 3 se probó que si HP $\in$ R entonces R = RE, demostrando que si existe una MT M_{HP} que decide HP, entonces para cualquier lenguaje L de la clase RE existe una MT M_L que lo decide. En realidad sólo se construyó M_L. Se pide probar que efectivamente M_L para siempre y que $L(M_L) = L$.

Ejercicio. Probar que la MT M_{20} construida en la clase 3 para decidir el lenguaje L_{20} = {<M> | M es una MT que a partir del input vacío λ nunca sale de las celdas 1 a 20}, efectivamente para siempre y acepta dicho lenguaje.

Ejercicio. Probar que la MT M_L construida en la clase 3 para aceptar L = {<M> | $L(M) \neq \emptyset$} efectivamente cumple que $L(M_L) = L$.

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

f(x) = 1, si x = (<M>, w) y M para a partir de w
f(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. En otras palabras, que se puede construir una MT que decide HP asumiendo que existe una MT que computa totalmente f_{HP}.*

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

Ejercicio. Probar el caso (b) del teorema presentado en clase, que enuncia:

Caso (a): Si L_1 α L_2 entonces $L_2 \in R \longrightarrow L_1 \in R$.
Caso (b): Si L_1 α L_2 entonces $L_2 \in RE \longrightarrow L_1 \in RE$.

Ayuda: basarse en la demostración del caso (a) desarrollada en clase.

Ejercicio. Sea el lenguaje $L_\emptyset$ = {<M> | $L(M) = \emptyset$}. Responder:

a. Encontrar una reducción de $L_U{}^C$ a $L_\emptyset$. *Ayuda: basarse en la idea de la reducción de L_U a L_{Σ^*}, es muy similar*.
b. Considerando la reducción desarrollada en (a), ¿qué se puede decir de $L_\emptyset$, a qué clase de la jerarquía de la computabilidad pertenece?

Ejercicio. Un autómata linealmente acotado (ALA) es una MT con una sola cinta con la restricción de que su cabezal sólo puede leer y escribir en las celdas en que se encuentra el input. Probar que el lenguaje aceptado por un ALA es recursivo. *Ayuda: ¿en cuántos pasos se puede detectar que el ALA entra en loop?*

Ejercicio. Probar mediante una reducción de problemas que L = {<M> | λ $\in$ L(M)} $\notin$ R, siendo λ la cadena vacía. *Ayuda: Basarse en alguno de los modelos de reducción vistos en clase.*

Ejercicio. Probar mediante una reducción de problemas que L = {<M> | L(M) = S, con S $\in$ RE y $S \neq \emptyset$} $\notin$ R. *Ayuda: Basarse en alguno de los modelos de reducción vistos en clase.*