TEORÍA DE LA COMPUTACIÓN Y VERIFICACIÓN DE PROGRAMAS 2025
Trabajo Práctico Nro 2

Indecibilidad. Reducciones (clases 3 y 4)

Ejercicio 1. Probar que el lenguaje L_U = {(<M>, w) | M acepta w} pertenece a la clase RE. *Comentario: probarlo construyendo una MT*.

Ejercicio 2. Responder breve y claramente cada uno de los siguientes incisos (en todos los casos, las MT mencionadas tienen una sola cinta):

a. Probar que 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 un loop?*
b. Probar que se puede decidir si una MT M que sólo se mueve a la derecha, a partir de una cadena w, para, *Ayuda: ¿Cuántos pasos puede hacer M antes de entrar en un loop?*
c. Probar que se puede decidir si dada una MT M, existe una cadena w a partir de la 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 una cadena w de a lo sumo 10 símbolos a partir de la cual M para? Justificar la respuesta.

Ejercicio 3. Sea M_1 una MT que genera un lenguaje recursivo L sin repetir cadenas y siguiendo el orden canónico. Probar que existe una MT M_2 que decide L. *Ayuda: cuidado con el caso en que L sea finito*.

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_2 generadas en los pares de salida (<M_2>, w), 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 la siguiente MT M_f no computa una reducción de HP a L_U: dada una cadena válida (<M>, w), M_f ejecuta M sobre w, si M acepta entonces genera la salida (<M>, w), y si M rechaza entonces genera la cadena 1.

Ejercicio 5. Sabiendo que $L_U \in$ RE y $L_U^C \in$ CO-RE:

a. Probamos en clase que existe una reducción de L_U a L_{Σ^*}. En base a esto, ¿qué se puede afirmar con respecto a la ubicación de L_{Σ^*} en la jerarquía de la computabilidad?
b. Se prueba que existe una reducción de L_U^C a $L_\emptyset$. En base a esto, ¿qué se puede afirmar con respecto a la ubicación de $L_\emptyset$ en la jerarquía de la computabilidad?

Ejercicio 6. Sea el lenguaje D_{HP} = {w_i | M_i para a partir de w_i} (considerar el orden canónico}. Encontrar una reducción de D_{HP} a HP. *Comentario: hay que definir la función de reducción y probar su total computabilidad y correctitud*.

Ejercicio 7. Sean TAUT y NOSAT los lenguajes de las fórmulas booleanas sin cuantificadores tautológicas (satisfactibles por todas las asignaciones de valores de verdad) e insatisfactibles (ninguna asignación de valores de verdad las satisface), respectivamente. Encontrar una reducción de TAUT a NOSAT. *Comentario: hay que definir la función de reducción y probar su total computabilidad y correctitud*.

Ejercicio 8. Se prueba que existe una reducción de L_U^C a L_{Σ^*} (y así, como $L_U^C \notin$ RE, entonces se cumple que $L_{\Sigma^*} \notin$ RE). La reducción es la siguiente. Para toda w: f((<M_1>, w)) = <M_2>, tal que M_2, a partir de su entrada v, ejecuta |v| pasos de M_1 a partir de w, y acepta sii M_1 no acepta. Probar que la función definida es efectivamente una reducción de L_U^C a L_{Σ^*} (es total computable y correcta).