TEORÍA DE LA COMPUTACIÓN Y VERIFICACIÓN DE PROGRAMAS
Trabajo Práctico Nro 1 Año 2021. Licenciatura en Informática.

Máquinas de Turing. Jerarquía o Mapa de la Computabilidad.

Comentario: Hacer los ejercicios del 1 al 5. El ejercicio 6 repasa algunas preguntas de la clase 2, que se recomiendan revisar pero cuyas respuestas no se piden entregar.

Ejercicio 1. Responder breve y claramente los siguientes incisos:

1. ¿Qué es un problema computacional de decisión? ¿Es el tipo de problema más general que se puede formular?
2. ¿Qué cadenas integran el lenguaje aceptado por una MT?
3. En la clase teórica 1 se hace referencia al problema de satisfactibilidad de las fórmulas booleanas. Formular las tres formas del problema, teniendo en cuenta las tres visiones de MT consideradas: calculadora, aceptadora o reconocedora, y generadora.
4. ¿Qué postula la Tesis de Church-Turing?
5. ¿Cuándo dos MT son equivalentes? ¿Cuándo dos modelos de MT son equivalentes?
6. ¿En qué difieren los lenguajes recursivos, recursivamente numerables no recursivos, y no recursivamente numerables?
7. Probar que $R \subseteq RE \subseteq \mathfrak{L}$.
8. ¿Todo lenguaje de la clase CO-RE tiene una MT que lo acepta?
9. Justificar por qué los lenguajes Σ^* y $\emptyset$ son recursivos.
10. Si $L \subseteq \Sigma^*$, ¿se cumple que $L \in R$?
11. Justificar por qué un lenguaje finito es recursivo.
12. Justificar por qué si $L_1 \in$ CO-RE y $L_2 \in$ CO-RE, entonces $(L_1 \cap L_2) \in$ CO-RE.
13. Dados $\Sigma = \{a, b, c\}$ y $L = \{a^n b^n c^n \mid n \geq 0\}$, obtener $\Sigma^* \cap L$, $\Sigma^* \cup L$, y el complemento de L respecto de Σ^*.

Ejercicio 2. Construir una MT, con cualquier cantidad de cintas, que acepte de la manera más eficiente posible el lenguaje $L = \{a^n b^n c^n \mid n \geq 0\}$. Plantear primero la idea general.

Ejercicio 3. Explicar (informal pero claramente) cómo simular una MT por otra que en un paso no pueda simultáneamente modificar un símbolo y moverse.

Ejercicio 4. Sean L_1 y L_2 dos lenguajes recursivamente numerables de números naturales representados en notación unaria (por ejemplo, el número 5 se representa con 11111). Probar que también es recursivamente numerable el lenguaje L = {x | x es un número natural representado en notación unaria, y existen y, z, tales que $y + z = x$, con $y \in L_1$, $z \in L_2$}.
Ayuda: la prueba es similar a la vista en clase de la clausura de RE respecto de la concatenación.

Ejercicio 5. Dada una MT M_1 con $\Sigma = \{0, 1\}$:

1. Construir una MT M_2 que determine si $L(M_1)$ tiene al menos una cadena.
2. ¿Se puede construir además una MT M_3 para determinar si $L(M_1)$ tiene a lo sumo una cadena? Justificar.

Ayuda para la parte (1): Si $L(M_1)$ tiene al menos una cadena, entonces existe al menos una cadena w de unos y ceros, de tamaño n, tal que M_1 a partir de w acepta en k pasos. Teniendo en cuenta esto, pensar cómo M_2 podría simular M_1 considerando todas las cadenas de unos y ceros hasta encontrar eventualmente una que M_1 acepte (¡cuidándose de los posibles loops de M_1!).

Ejercicio 6. Considerando los lemas estudiados en la clase 2, indicar/probar/construir:

1. Cómo se implementaría copiar el input w en la cinta 2 de la MT M (Lema 2).
2. Cómo se implementaría borrar el contenido de la cinta 2 de la MT M (Lema 2).
3. Probar la correctitud de la construcción (Lema 2).
4. Cómo se implementaría la suma de 1 al contador i en la MT M (Lema 3).
5. Cómo se implementaría ejecutar en M, i pasos de las MT M_1 y M_2 (Lema 3).
6. Probar la correctitud de la construcción (Lema 3).
7. Construir la MT M solución y probar su correctitud (Lema 4).
8. Probar la correctitud de las construcciones de las MT de la Clase Práctica 2.
9. Probar las otras propiedades de clausura de R y RE mencionadas en la Clase 2.