Máquinas de Turing
Jerarquía de la Computabilidad.

March 27, 2019

Agustin Vanzato
Federio Gasquez

1. **Responder brevemente los siguientes incisos:**

   (a) **¿Qué es un problema (computacional) de decisión? ¿Es el tipo de problema más general que se puede formular?** Un problema de decisión es un problema en donde las respuestas posibles son "sí" o "no".

   (b) **¿Qué cadenas integran el lenguaje aceptado por una MT?** El conjunto de las cadenas aceptadas por la MT M es el lenguaje aceptado o reconocido por M, y se denota con L(M). Considerando la visión de MT M calculadora, sólo cuando M se detiene en un estado q ∈ F debe tenerse en cuenta el contenido final de la cinta, es decir la cadena de salida (o simplemente la salida).

   (c) **En la clase teórica 1 se hace referencia al problema de satisfactibilidad de las fórmulas booleanas (se da como ejemplo la fórmula $\varphi = (x_1 \vee x_2) \wedge (x_3 \vee x_4)$ y la asignación A = (V, F, V, V)).**
   Formular las tres formas del problema, teniendo en cuenta las tres visiones de MT consideradas: calculadora, aceptadora o reconocedora y generadora.

   (d) **¿Qué postula la Tesis de Church-Turing?**
   la conjetura conocida como Tesis de Church-Turing, indica que todo lo computable puede ser llevado a cabo por una máquina de Turing

   (e) **¿Cuándo dos MT son equivalentes? ¿Cuándo dos modelos de MT son equivalentes?** Dos máquinas de turing son equivalentes cuando reconocen el mismo lenguaje. Dos modelos de máquina de turing son equivalentes cuando poseen el mismo poder de cómputo, es decir a partir de una se puede simular la otra y viceversa.

(f) **¿En qué difiere un lenguaje recursivo de un lenguaje recursivamente numerable no recursivo?**

Un lenguaje es recursivamente numerable si y sólo si existe una MT que lo reconoce. Es decir, si L es el conjunto de todos los lenguajes (cada uno integrado por cadenas finitas de símbolos pertenecientes a un alfabeto universal $\sum$), sólo los lenguajes recursivamente numerables de L son reconocibles por una MT (por esto es que a los problemas de decisión asociados se los conoce como computables). La clase de los lenguajes recursivamente numerables se denomina RE (por recursively enumerable languages). El nombre se debe a que las cadenas de estos lenguajes se pueden enumerar. De esta manera, dado L $\in$ RE, si M es una MT tal que L(M) = L, se cumple para toda cadena w de $\sum^*$ que:

- Si w $\in$ L, entonces M a partir de w se detiene en su estado q_A .
- Si w $\notin$ L, entonces M a partir de w se detiene en su estado q R o no se detiene.

Se define que un lenguaje es recursivo si y sólo si existe una MT M que lo reconoce y que se detiene cualquiera sea su entrada. La clase de los lenguajes recursivos se denomina R. A los problemas de decisión asociados se los conoce como decidibles, porque las MT que los resuelven pueden justamente decidir, cualquiera sea la instancia, si es positiva o negativa. Ahora, dado L $\in$ R, si M es una MT tal que L(M) = L, se cumple para toda cadena w de $\sum^*$ que:

- Si w $\in$ L, entonces M a partir de w se detiene en su estado q A .
- Si w $\notin$ L, entonces M a partir de w se detiene en su estado q R .

(g) **¿En qué difiere un lenguaje recursivamente numerable de uno que no lo es?**

(h) **Probar que R $\subseteq RE \subseteq \mathcal{L}$.**

(i) **¿Cuándo un lenguaje está en la clase CO-RE? ¿Puede un lenguaje estar al mismo tiempo en la clase RE y en la clase CO-RE? ¿Para todo lenguaje de la clase CO-RE existe una MT que lo acepta?**

(j) **Justificar por qué los lenguajes $\sum$ y $\emptyset$ son recursivos.**

(k) **Si $L \subseteq \sum^*$, ¿se cumple que L $\in$ R?**

(l) **Justificar por qué un lenguaje finito es recursivo.**

(m) **Justificar por qué si L1 $\in$ CO-RE y L2 $\in$ CO-RE, entonces (L1 $\wedge$ L2) $\in$ CO-RE**

2. **Dado el alfabeto $\sum = \{$a, b, c$\}$:**

(a) **Obtener el lenguaje $\sum^*$ y el conjunto de partes del subconjunto de $\sum^*$ con cadenas de a lo sumo dos símbolos. ¿Cuál es el cardinal (o tamaño) de este último conjunto?**

(b) **Dado el lenguaje $L = a^n b^n c^n | n \geq 0$, obtener la intersección $\sum^* \cap$ L, la unión $\sum^* \cup$ L, el complemento de L respecto de $\sum^*$, y la concatenación $\sum^*$. L.**

3. **Construir una MT (puede tener varias cintas) que acepte de la manera más eficiente posible el lenguaje $L = \{a^n b^n c^n | n \geq 0\}$. Plantear primero la idea general.**

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

5. **Explicar (informal pero claramente) cómo simular una MT por otra que no tenga el movimiento S (es decir el no movimiento).**

6. **Sea USAT el lenguaje de las fórmulas booleanas satisfactibles con exactamente una asignación de valores de verdad. P.ej. $x_1 \wedge x_2$ pertenece a USAT, mientras que $(x_1 \wedge x_2) \vee x_3$ no. Indicar, justificando la respuesta, si la siguiente MTN acepta USAT:**

   (a) **Si la fórmula de entrada no es correcta sintácticamente, rechaza.**

   (b) **Genera no determinísticamente una asignación A, y si A no satisface la fórmula, rechaza.**

   (c) **Genera no determinísticamente una asignación A' $\neq$ A. Si A' no satisface la fórmula, acepta, y si A' la satisface, rechaza.**

   Ayuda: Considerar p.ej. el caso en que la fórmula tiene dos asignaciones que la satisfacen.

7. **Considerando el Lema 4 estudiado en la Clase Teórica 2** ($R = RE \cap CO - RE$):

   (a) **Construir la MT M.**

   (b) **Probar la correctitud de la construcción.**

8. **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 de la clausura de RE con respecto a la concatenación.

9. **Dada una MT M1 con $\sum =$**

   (a) **Construir una MT M_2 que determine si L(M_1) tiene al menos una cadena.**

   (b) **¿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(M1) tiene al menos una cadena, entonces existe al menos una cadena w de unos y ceros, de tamaño n, tal que M1 a partir de w acepta en k pasos. Teniendo en cuenta esto, pensar cómo M2 podría simular M1 considerando todas las cadenas de unos y ceros hasta encontrar eventualmente una que M1 acepte (¡cuidándose de los casos en que M1 entre en loop!)