

# Teoría Matemática de la Computación

## Primer Problemario

Prof. Miguel A. Pizaña  
22 de Febrero de 2007

### I Tareas

1. Dudar de todo, al menos una vez en la vida.
2. ¿Qué emociones le produce el teorema de Cantor, la inexistencia del conjunto universo, la irresolubilidad de la mayoría de los problemas, la irresolubilidad del problema del Paro? (esta tarea es para entregar).
3. ¿Quién es Noam Chomsky?
4. Averiguar cuál es el máximo nivel de anidación que permite su compilador favorito.
5. Leer *On Computable Numbers, with an Application to the Entscheidungsproblem* de Alan Turing. Luego responda: ¿Cómo le llama Turing a sus máquinas? ¿Qué diferencias hay entre su definición y la definición vista en clase? ¿Qué es para Turing un número computable? ¿son computables  $\pi$ ,  $\sqrt{2}$ ,  $\frac{1}{3}$ ,  $\tan(23)$ ? ¿Cuál es la razón por la que la máquina usa una cinta para hacer anotaciones? ¿Por qué usa un número finito de símbolos? ¿Qué es el Entscheidungsproblem? ¿Qué prueba Turing al respecto?
6. Leer *Computer Machinery and Intelligence* de Alan Turing. Luego conteste: ¿Qué es la prueba de Turing (Turing's Test)? ¿Usted cree que alguna máquina podrá algún día aprobar la prueba de Turing? Suponiendo que sí, ¿Eso indicaría que la máquina tiene inteligencia? ¿Puede el cerebro humano hacer algo que las Máquinas de Turing no? Según Turing, ¿pueden las máquinas pensar? ¿Qué argumentos en favor y en contra da Turing? Según usted, ¿pueden las máquinas pensar? ¿por qué?

7. Muy importante: ¿Qué dice el artículo 39 de la Constitución Política de los Estados Unidos Mexicanos?

## II Modelos vs Realidad

1. Explique las diferencias y similitudes entre modelo (matemático) y realidad.
2. ¿Cómo hace uno para medir qué tan bien aproxima un modelo matemático a la realidad?
3. La “ley de conservación de la materia y la energía”, ¿Es parte de un modelo, o parte de la realidad? ¿Qué tan cierta es esta ley?
4. ¿Qué tan cierto es el teorema de Pitágoras? Explique con detalle. Proporcione una prueba del teorema de Pitágoras.
5. Muestre que el teorema de Pitágoras y la idea de una longitud mínima (= longitud de Planck) son mutuamente contradictorias. ¿Cuál de ellas es verdadera? Explique.
6. ¿Cuál de las siguientes afirmaciones es verdadera?
  - (a) La siguiente afirmación es falsa.
  - (b) La afirmación anterior es verdadera.
  - (c) Las dos afirmaciones anteriores son falsas.
  - (d) Las dos primeras afirmaciones son verdaderas.
  - (e) Ninguna de las anteriores es verdadera.
7. ¿Es verdadera la afirmación: “Esta afirmación es falsa”?
8. ¿Cuál es el más pequeño de los números reales positivos?
9. Suponga que lanza usted una moneda un millón de veces y en todas esas ocasiones usted ve a la moneda caer hasta el suelo en un lindo arco parabólico. ¿Eso es garantía de que la siguiente vez que usted lance la moneda, ésta no se va a convertir en un monstruo de espagueti volador? Explique.
10. Suponga que usted juega al *melate* un millón de veces y en todas esas ocasiones usted NO se saca el premio. ¿Eso es garantía de que la siguiente vez que usted juegue *melate* tampoco se va a sacar el premio? Explique.

11. Pruebe el teorema de Schröder-Bernstein: Para cualesquiera dos conjuntos  $A$  y  $B$  tenemos que  $|A| \leq |B|$  y  $|B| \leq |A|$  implican  $|A| = |B|$
12. Pruebe el teorema de Cantor: para todo conjunto  $X$ , tenemos que  $|X| < |\mathcal{P}(X)|$ .
13. Pruebe que el universo no existe.
14. Suponga que  $\Sigma$  es cualquier conjunto finito no vacío. Pruebe que:  $|\mathbb{N}| = |\mathbb{N} \times \mathbb{N}| = |\mathbb{Q}| = |\mathbb{Q} \times \mathbb{Q}| = |\Sigma^*| = |\{\text{programas}\}|$ .
15. Pruebe que:  $|\mathbb{R}| = |(0, 1)| = |[0, 1]| = |\mathbb{C}| = |\mathbb{R}^n| = |\mathcal{P}(\mathbb{N})| = |\mathcal{P}(\Sigma^*)| = |\{\text{problemas}\}|$ .
16. Pruebe que  $|\mathbb{N}| < |\mathbb{R}|$ .
17. Pruebe que la mayoría de los problemas son irresolubles.
18. Pruebe que el problema del paro es irresoluble. Pruebe también que es irresoluble el problema de decidir si un programa dado  $P$ , con una entrada dada  $E$ :
  - (a) Va alguna vez a escribir algo en pantalla.
  - (b) Va alguna vez a acceder al disco duro.
  - (c) Va a enviar un email.
  - (d) Va a formatear el disco duro.
  - (e) Es viral (va a hacer copias de sí mismo).
19. ¿Cuántos números reales hay? ¿Cuántos de ellos pueden ser representados en una computadora real (de memoria finita)? Si las computadoras pudieran tener memoria ilimitada (=la memoria es infinita, pero sólo podemos usar una cantidad finita de casillas en cada momento) ¿Cuántos números reales podríamos representar en la computadora? ¿Cómo es que trabajamos con las computadoras con los números de punto flotante como si todo esto no representara un problema?
20. “Sumar uno” es una operación que no se puede hacer con memoria acotada. ¿Cómo es que usamos las computadoras para sumar uno?
21. ¿Qué dice el principio del buen orden?
22. Sea  $H$  el conjunto de todos los números reales que pueden ser definidos por un humano (con un número finito de símbolos, por

ejemplo). Claramente  $|H| = \aleph_0$ , pero  $|\mathbb{R}| = c$ , de modo que  $\mathbb{R} - H$  no es vacío. Sea ahora  $x$  el más pequeño (de acuerdo a algún buen orden de los reales) de entre los números reales en  $\mathbb{R} - H$ . Acabo de definir un número real fuera de  $H$ ... entonces, ¿no soy humano?

23. ¿Cuáles son los 3 principales modelos matemáticos que se usan para aproximar las nociones de “algoritmo” y “computadora”? ¿Cuáles son las principales características de estos modelos? ¿Qué ventajas y desventajas relativas tienen? ¿Para qué se usa cada uno de estos modelos?

### III Máquinas de Acceso Aleatorio.

1. ¿Qué es un a Máquina de Acceso Aleatorio? ¿Cómo se define? ¿Como se llama la teoría que emerge del estudio de este modelo? ¿Cuáles son sus principales características? ¿para que sirve? ¿Cuánto tiempo tarda en realizar un operación elemental? ¿Qué almacena en cada celda de memoria? ¿cuántas celdas tiene? ¿Cuáles son las principales variaciones del modelo RAM básico? ¿Cuáles aspectos de la realidad aproxima bien este modelo? ¿Cuáles aproxima mal? ¿Cuántos y cuáles son los supuestos falsos en los que se basa este modelo? ¿Qué paradojas produce este modelo de la computación?
2. De un algoritmo para el problema de la satisfacibilidad (SAT) que funcione en tiempo polinomial en el modelo RAM de costo unitario. ¿Porqué este algoritmo no vale un millón de dólares?
3. Muestre que el algoritmo usual para multiplicar matrices tarda un tiempo del orden de  $n^3$ .
4. El mejor algoritmo conocido para calcular matrices tarda un tiempo del orden de  $n^\alpha$ . ¿Cuánto vale  $\alpha$ ? ¿Quién o quienes descubrieron ese algoritmo?
5. Muestre que es físicamente imposible tener una computadora con memoria arbitrariamente grande (aunque finita) en donde se pueda tener acceso a cualquiera de sus celdas en menos de un milisegundo.
6. La “Ley de Moore” afirma que la cantidad de transistores que se pueden poner en un chip se duplica cada 18 meses. Muestre que es físicamente imposible que esto pueda ocurrir por siempre.

#### IV Máquinas de Turing.

1. ¿Qué es una Máquina de Turing? ¿Cómo se define? ¿Cómo se llama la teoría que emerge del estudio de este modelo? ¿Cuáles son sus principales características? ¿para que sirve? ¿Cuánta memoria tiene? ¿Qué operaciones básicas sabe realizar una MT? ¿Cuáles aspectos de la realidad aproxima bien este modelo? ¿Cuáles aproxima mal? ¿Cuántos y cuáles son los supuestos falsos en los que se basa este modelo?
2. ¿Qué dice la tesis de Church-Turing? ¿Cree usted en la tesis de Church-Turing? ¿Puede esta tesis ser demostrada matemáticamente? Explique.
3. En cada inciso proporcione una Máquina de Turing que:
  - (a) Inserte un símbolo al inicio de la cinta.
  - (b) Dado un número binario no negativo  $n$ , calcule  $n + 1$ .
  - (c) Dado un número binario positivo  $n$ , calcule  $n - 1$ .
  - (d) Dado un par de número binarios separados por un “#”, calcule la suma.
4. Para cada uno de los siguientes lenguajes proporcione una máquina de Turing (que siempre para) que los reconozca:
  - (a)  $\{n : n \text{ es par} \}$
  - (b)  $\{0^n 1^n : n \in \mathbb{N}\}$ .
  - (c)  $\{ \text{paréntesis balanceados} \}$ .
  - (d)  $\{ \text{expresiones aritméticas con paréntesis} \}$
  - (e)  $\{ \text{palíndromos de 0's y 1's} \}$ .
  - (f)  $L_m = \{0^{n+m} 1^n : n \geq 0\} \cup \{0^n 1^{n+m} : n \geq 0\}$ .
  - (g)  $\{w \in \{0, 1\}^* : w \text{ tiene igual número de 0's que de 1's} \}$ .
5. Demuestre que todo lo que un AFD puede hacer, también lo puede hacer alguna máquina de Turing.
6. ¿Qué es una Máquina de Turing No Determinista? Si  $P = \text{polinomial}$ , ¿que es NP? ¿Es cierto que  $P = NP$ ? ¿Es cierto que  $P \cap NP = \emptyset$ ?

#### V Autómatas Finitos Deterministas

1. ¿Qué es un Autómata Finito Determinista? ¿Cómo se define? ¿Cuáles son sus principales características? ¿Para qué sirve? ¿Cómo se define el *lenguaje reconocido* por un AFD? ¿Cuáles aspectos de la realidad aproxima bien este modelo? ¿Cuáles aproxima mal?
2. Para cada uno de los siguientes lenguajes, proporcione un AFD que los reconozca (suponga que  $\Sigma = \{0, 1\}$ ).
  - (a) Cadenas que terminen en 00.
  - (b) Cadenas que contengan tres (o más) ceros consecutivos.
  - (c) Cadenas que contengan tres (pero no más de tres) ceros consecutivos.
  - (d) Cadenas que contengan la subcadena 000 exactamente una vez.
  - (e) Cadenas que representen números enteros divisibles entre 3.
  - (f) Cadenas que representen números enteros divisibles entre 7.
  - (g) Cadenas que tengan un 0 en la décima posición contando desde la izquierda.
  - (h) Lo mismo que el anterior, pero contando desde la derecha.
  - (i) Cadenas que contengan a 0101 como subcadena.
  - (j) Cadenas que NO contengan a 0101 como subcadena.
  - (k) Cadenas tal que en cada subcadena de longitud 5 haya al menos 2 ceros.
  - (l) Cadenas tal que en cada prefijo el valor absoluto de la diferencia entre el número de 1's y el número de 0's sea menor o igual a 5.
  - (m) Cadenas como en el inciso anterior en donde además el número total de ceros es igual al número total de unos.
3. Haga un lista completa de todos los AFD con alfabeto binario que tengan uno o dos estados (son 66, pero muchos de ellos son equivalentes). ¿Qué lenguaje reconoce cada uno de ellos?
4. Demuestre que si permitimos que el conjunto de estados en un Autómata Determinista sea infinito, entonces, dado cualquier lenguaje  $L \subseteq \Sigma^*$  existe un Autómata Determinista (posiblemente infinito)  $M$  tal que  $L = L(M)$ .
5. Demuestre que dado cualquier lenguaje **finito**  $L$  existe un AFD que lo reconoce.

6. ¿Existe un AFD que dada cualquier matriz de  $n \times n$  ( $n < 50$ ) de números de punto flotante de doble precisión (todo ello codificado como una cadena binaria) decida si es invertible? Explique.
7. Los AFD sólo modelan algoritmos de decisión. Desarrolle usted su propio modelo matemático basado en el del AFD (memoria finita) que modele **cualquier** procesamiento de información  $f : \Sigma^* \rightarrow \Sigma^*$  “computable con memoria finita”. Por ejemplo, su modelo debería poder transformar cualquier cadena  $w \in \Sigma^*$  de ceros y unos en la cadena  $w'$  que consta del doble de unos que  $w$  y no tiene ceros.
8. Modifique su modelo anterior para que sea capaz de percibir el paso del tiempo y actuar en respuesta a ello. Por ejemplo, su modelo debería poder mostrar el comportamiento del teclado: Si oprimes y sueltas (rápidamente) una tecla aparece la correspondiente letra en pantalla, pero si oprimes y mantienes oprimida la tecla aparecerá la tecla correspondiente con un número de repeticiones que depende del tiempo transcurrido.
9. ¿Puede usted hacer un programa de computadora (con memoria finita) que lea un número natural  $n$  expresado en binario y regrese  $n+1$  también expresado en binario? ¿Y si el número debe ser leído y escrito al revés (es decir en formato *bigendian*)? ¿Puede hacer un programa que regrese  $2n$ ? ¿ $2n+1$ ? ¿ $n^2$ ?