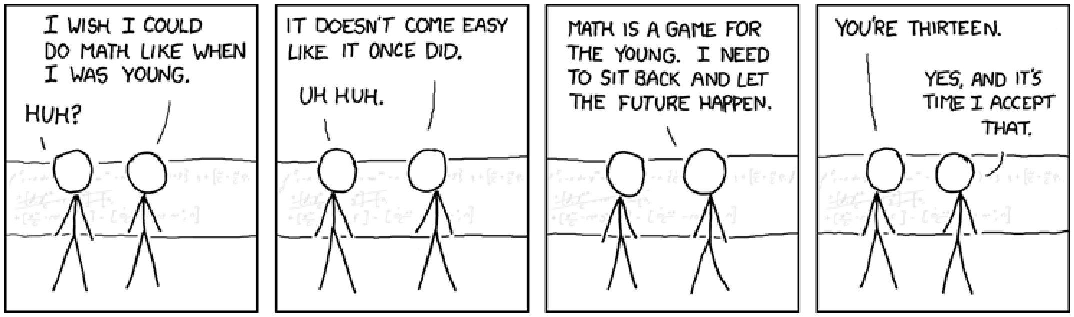
Segundo Certamen
Introducción a la Informática Teórica

19 de julio de 2014



1. Explique (mediante una construcción informal, pero convincente) sus respuestas a las siguientes:

   a) ¿Son cerrados respecto a unión los lenguajes recursivamente enumerables?

   b) ¿Son cerrados respecto a complemento los lenguajes recursivamente enumerables?

   c) ¿Son cerrados respecto de intersección los lenguajes recursivos?

   (30 puntos)

2. Una descripción de máquina de Turing es que consta del *alfabeto de cinta* Γ (incluye Σ, el alfabeto de entrada, y el símbolo blanco $b \notin \Sigma$), el conjunto finito Q de *estados* que incluye q_0 (inicial), q_Y (acepta) y q_N (rechaza); y una *función de transición* $\delta \colon (Q \smallsetminus \{q_Y, q_N\}) \times \Gamma \to Q \times \Gamma \times \{-1, +1\}$. La *entrada* es $\sigma \in \Sigma^*$, en posiciones 1 a $|\sigma|$ de la cinta, las demás contienen b. Partiendo del ejemplo $\sigma = 10100$ determine el lenguaje que acepta la máquina $\Gamma = \{0, 1, b\}$, $\Sigma = \{0, 1\}$, $Q = \{q_0, q_1, q_2, q_3, q_Y, q_N\}$ y δ dada por la tabla:

q	**0**	**1**	b
q_0	$(q_0, 0, +1)$	$(q_0, 1, +1)$	$(q_1.b, -1)$
q_1	$(q_2, b, -1)$	$(q_3, b, -1)$	$(q_N, b, -1)$
q_2	$(q_Y, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$
q_3	$(q_N, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$

   (15 puntos)

3. Demuestre que no hay un algoritmo que determine si dos programas entregan los mismos resultados.

   (25 puntos)

4. El problema de *scheduling* de una máquina en el cual tenemos r tareas, la tarea i con tiempo de ejecución p_i y plazo fatal d_i, donde se busca minimizar el número de tareas atrasadas (que se completan después de su plazo fatal d_i) puede resolverse mediante el método de Moore:

   - Se agregan tareas al final de la programación actual en orden de plazo fatal creciente.
   - Si al agregar la tarea i se sobrepasa su plazo fatal, se elimina de la programación una de las tareas con máximo tiempo de ejecución, y se declara retrasada.
   - Una vez consideradas todas las tareas, las retrasadas se programan en orden arbitrario al final.

   Demuestre que este algoritmo se ejecuta en tiempo polinomial, esbozando estructuras de datos y algoritmos empleados.

   (15 puntos)

5. El problema JOB SEQUENCING, descrito mediante r tareas de tiempos de ejecución $(t_1, \dots, t_r) \in \mathbb{Z}^r$, plazos fatales $(d_1, \dots, d_r) \in \mathbb{Z}^r$ y penalidades $(p_1, \dots, p_r) \in \mathbb{Z}$ donde se pregunta si hay una secuencia de ejecución de las tareas que minimiza la suma de las penalidades de las tareas que terminaron luego de su plazo fatal es el problema 19 en la lista de Karp de problemas NP-completos. Explique la aparente contradicción con la conclusión del problema 4, que describe un caso particular.

(20 puntos)

6. Usando los problemas NP-completos vistos en clase o tareas, demuestre formalmente que el problema SUBGRAPH ISOMPORPHISM (¿Contiene G un grafo isomorfo a G' como subgrafo?) es NP-completo

(20 puntos)