CURSO DE COMPLEJIDAD COMPUTACIONAL 2021

Obligatorio 1

1. Un autómata celular unidimensional es un vector lineal de procesadores que se extiende arbitrariamente lejos en ambas direcciones, y que operan en forma síncrona. Cada procesador tiene el mismo conjunto finito de estados Q, y en cada paso de tiempo cada procesador cambia su estado desde su estado actual q a un nuevo estado dado por $\delta(p,q,r)$, donde p es el estado del procesador situado a su izquierda y r el estado del procesador ubicado a su derecha. Uno de los estados es el estado de parada (halt), y el autómata completo se detiene cuando *cualquiera* de sus procesadores entra en el estado de parada. El autómata se inicializa con una tira binaria $w = w_1 ... w_n$ colocando cada elemento w_i en el procesador i. El estado inicial del procesador i es 0 si $w_i = 0$, es 1 si $w_i = 1$ y vale 2 si dicho procesador no tiene asignado un w_i de la entrada. Se asume que $\delta(2,2,2) = 2$ (un procesador del autómata celular cuyos procesadores vecinos no fueron influenciados todavía por la entrada mantiene su estado inicial incambiado).

a) Sea un autómata celular unidimensional M_u con entrada w.
Mostrar que existe una una MT multicinta convencional *M* (modelo Arora-Barak) que simula exactamente el comportamiento de M_u con entrada w. En particular luego de cada paso de ejecución de M_u, la MT *M* debe llegar a una configuración que contenga el estado de cada procesador de M_u al final de dicho paso de M_u. Detallar completamente la estructura y el funcionamiento de *M.*

b) Para una entrada w con $|\mathrm{w}| = n$, determinar una expresión para el orden del tiempo de ejecución acumulado de la MT correspondiente a k pasos de tiempo de la ejecución del aútomata celular unidimensional M_u. Mostrar que si para dicha entrada de tamaño n el autómata M_u para en $p(n)$ pasos, donde p es un polinomio, la MT correspondiente para en $O(q(n))$ pasos, siendo q otro polinomio. Expresar $q(n)$ en función de $p(n)$.

c) Si L es un lenguaje en P, mostrar que existe un autómata celular unidimensional M_u tal que para cualquier entrada w con $|\mathrm{w}| = n$, M_u para luego de una cantidad $r(n)$ de pasos, siendo r un polinomio, y el estado del procesador 0 cuando M_u para será 0 si $\mathrm{w} \notin L$ y 1 si $\mathrm{w} \in L$. Describir en detalle el funcionamiento de M_u, especificando en particular que información se codifica en cada estado de los procesadores de M_u y como se define la función de transición $\delta(p,q,r)$.

2. Se considera la reducción polinomial de Karp entre lenguajes denotada con $\leq_P$ y según definición 2.7 del texto "Computational Complexity" de Arora-Barak.
Mostrar que no es una relación **simétrica**, es decir que si $L \leq_P L^*$ no necesariamiente se cumple que $L^* \leq_P L$.

3. Se consideran dos lenguajes L y L^*.
Se sabe que $L^* \in NP$ y que existe una reducción polinomial de Karp $L \leq_P L^*$.
Probar que $L \in NP$.

4. Consideramos el problema
$CNF_k = \{\phi \mid \phi \text{ es una fórmula } CNF \text{ satisfactible en donde cada variable aparece a lo sumo } k \text{ veces}\}$
a) Probar que CNF_3 es NP-completo.
b) Se sabe que $CNF_2 \in P$. Explicar en qué falla su prueba de a) al aplicarla para demostrar que CNF_2 es NP-completo.

5. Se considera una instancia del problema de satisfactibilidad SAT, especificado por un conjunto de cláusulas sobre un conjunto de variables booleanas $x_1,...,x_n$. Se dice que una instancia es <u>*monótona*</u> si cada término (literal) de cada cláusula consiste en una variable no negada, esto es cada literal es igual a x_i, para algún i, y no aparecen literales del tipo $\overline{x_i}$. Las instancias monótonas de SAT son siempre satisfactibles asignándole valor de verdad 1 a cada variable. Consideremos por ejemplo el conjunto de cláusulas: $\{(x_1 \vee x_2),(x_1 \vee x_3),(x_2 \vee x_3)\}$.
Esta instancia de SAT es monótona, y la asignación de 1 a todas las variables satisface todas las cláusulas. Pero se observa que se podrían haber seteado alternativamente los valores de x_1 y x_2 en 1 y x_3 en 0 y seguir satisfaciendo las tres cláusulas. Resulta entonces que para cualquier instancia monótona es natural preguntarse cual es el número <u>mínimo</u> de variables x_i que es necesario fijar en 1 para satisfacer dicha instancia.
Dada una instancia monótona de SAT y un número k, el problema de Satisfactibilidad Monótona con Pocas Variables Verdaderas (SAT-MON) en su versión de decisión pregunta: ¿Hay alguna asignación de valores de verdad a las variables que satisfaga la instancia y en la cual <u>a lo sumo</u> k variables se fijen en valor 1? . Probar que SAT-MON es NP-completo.

6. Se tiene que definir un calendario para un conjunto $E = \{E_1,...,E_k\}$ de exámenes que pueden ser tomados por estudiantes del conjunto $S = \{S_1...S_m\}$. Cada estudiante va a rendir un subconjunto específico de estos exámenes. Se desea programar los exámenes de forma tal que ningún estudiante tenga que tomar más de un exámen en un mismo día.
Se define $SCHED$ como el siguiente problema de decisión: dado un entero h, ¿existe una programación para los exámenes que requiera a lo sumo h días?.
Sea por otra parte el problema $\overline{INDSET}$: dado un grafo no dirigido G y un entero k, ¿se verifica que G **<u>no</u>** tiene un conjunto independiente de nodos de tamaño mayor o igual a k ?
Alguien afirma que existe una reducción de tiempo polinomial del problema $\overline{INDSET}$ al problema $SCHED$, es decir que $\overline{INDSET} \leq_P SCHED$. La función de mapeo f propuesta es la siguiente: dado el par (G,k) de la instancia de $\overline{INDSET}$, se identifica cada vértice u de G

con un examen E_u de la instancia correspondiente de $SCHED$. Si no hay arista entre un par de vértices u y v de G, se incluye un estudiante S_{uv} que debe rendir ambos exámenes E_u y E_v. Se toma $h = k - 1$.

a) ¿Es posible implementar la función f en tiempo polinomial en el tamaño de la instancia? Justificar.

b) Sea x una instancia del problema $\overline{INDSET}$ cuya transformada $f(x)$ es una instancia del problema $SCHED$ que tiene respuesta afirmativa (instancia SI de $SCHED$). ¿Se cumple necesariamente que x es una instancia SI de $\overline{INDSET}$? . Probar o dar un contraejemplo.

c) Determinar si la reducción propuesta es una reducción de Karp válida.

7. Probar que $\mathrm{NP{=}coNP}$ si y solo si 3SAT y TAUTOLOGY son polinomialmente reducibles el uno al otro.

8. Dar un ejemplo de un lenguaje R que verifique al mismo tiempo:

a) $R \in NP$

b) $R \in coNP$

c) No se sabe si $R \in P$

Justifique en detalle el cumplimiento de las propiedades a) y b) .