Universidad Nacional Autónoma de México
Facultad de Ciencias
Complejidad Computacional

Tarea 3

Ángel Iván Gladín García
No. cuenta: 313112470
angelgladin@ciencias.unam.mx

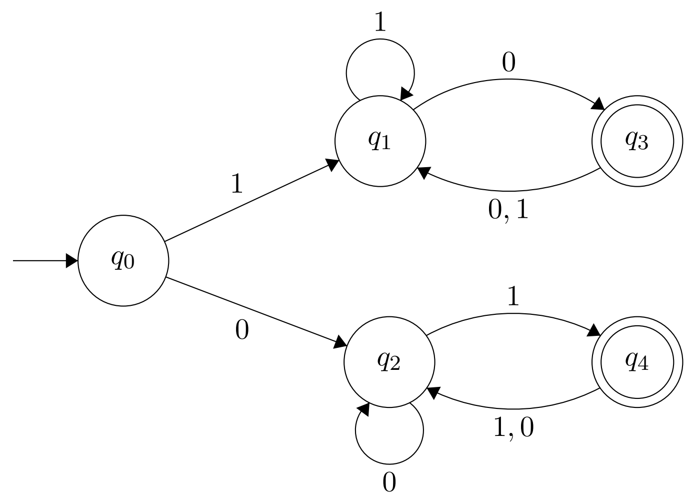
17 de Septiembre 2019

1. Seleccione **uno** de los siguientes dos ejerccios:

   a) Defina un *programa* **M** para una MTD que acepte el siguiente lenguaje:

   $$\mathscr{L} = \{0(0+1)^*0 + 1(0+1)^*1\}$$

   Lo primero que se hará por cuestiones de facilidad es crear un autómata finito no determinista *(NFA)* y después pasarlo a una MTD.

   Recordando la definición de MT que es:

   $$M = (Q, \Sigma, \Gamma, \delta, s, B, F)$$

   Donde tenemos que:

   - $Q = \{q_0, q_1, q_2, q_3, q_4\}$
   - $\Sigma = \{0, 1\}$
   - $\Gamma = \{0, 1, \#\}$
   - δ dada por:

Estado	0	1	#
q_0	$(q_1, 1, R)$	$(q_2, 1, R)$	$-$
q_1	$(q_3, 0, R)$	$(q_1, 1, R)$	$-$
q_2	$(q_2, 0, R)$	$(q_4, 1, R)$	$-$
q_3	$(q_1, 0, R)$	$(q_1, 1, R)$	$(q_3, \#, R)$
q_4	$(q_2, 0, R)$	$(q_2, 1, R)$	$(q_4, \#, R)$

   - $s = q_0$
   - $B = \#$
   - $F = \{q_3, q_4\}$

   b) Defina un *programa* **M** para una MTD que acepte el siguiente lenguaje:

   $$\mathscr{L} = \{1^n 0^m | n, m \geq 0 \wedge 2|(n+m)\}$$

   Hint:

   $$\mathscr{L} = \{(11)^*(00)^* + 1(11)^*0(00)^*\}$$

2. Definir formalmente, en términos de problemas y en términos de lenguajes, las clases:

   - *CoP*
     - En términos de problemas:

       $$\text{co-P} = \{\Pi^c : \Pi \in P\}$$

     - En términos de lenguajes:
       Sea Π un problema en P codificado en un alfabeto Σ y decimos que hay una cadena s que es aceptado, por otro lado, sea Π^c un problema en P decimos que hay una cadena w que es aceptada.
   - *CoNP*
     - En términos de problemas:

       $$\text{co-NP} = \{\Pi^c : \Pi \in NP\}$$

     - En términos de lenguajes:

       $$\text{co-NP} = \{\Sigma^* - L : L \text{ es es una lenguaje sobre el alfabeto } \Sigma \text{ y } L \in NP\}$$

3. Seleccionar dos de los siguientes problemas en la clase **P**, plantearlos como problemas de decisión y enunciar sus complementos.

   - Problema Flujo Máximo
   - Problema de Ruta más corta.
     Ejemplar: Dada un gráfica $G = (V, E)$, un peso en cada arista $w(e)$, y un número k.
     Pregunta: ¿Existe un camino de u a v de a lo más k (la suma del peso de sus aristas)?

     Complemento:
     Ejemplar: Dada un gráfica $G = (V, E)$, un peso en cada arista $w(e)$, y un número k.
     Pregunta: ¿Para todo camino de u a v es mayor que k?
   - Problema Apareamiento en gráficas bipartitas
     Ejemplar: Dada una gráfica bipartita $G = (U, V, E)$, un apareamiento M y un número k.
     Pregunta: ¿Existe un apareamiento bipartito que sea por lo menos de tamaño k?

     Complemento:
     Ejemplar: Dada una gráfica bipartita $G = (U, V, E)$, un apareamiento M y un número k.
     Pregunta: ¿Para todo apareamiento bipartito es de tamaño menor a k?

4. Seleccionar dos de los siguientes problemas en la clase **NP**, plantearlos como problemas de decisión y enunciar sus complementos.

   - Problema de Coloración en Gráficas.
   - Problema del Clan:
     Ejemplar: Gráfica G, entero positivo K.
     Pregunta: ¿La subgráfica completa más grande en G contiene exactamente K vértices?

     Complemento:
     Ejemplar: Gráfica G, entero positivo K.
     Pregunta: ¿K es el tamaño del clan máximo?

- Problema Conjunto Independiente
  Ejemplar: Sea una gráfica $G = (V, E)$ no dirigida y sea $I \subseteq V$. Decimps que I es un conjunto independiente si cualesquirra $i, j \in I$ entonces no hya una arista entre i y j
  Pregunta: Sea una gráfica $G = (V, E)$ no dirigida y una meta k, ¿hay un conjunto independiente I con $|I| = k$?

  Complemento:
  Ejemplar: Sea una gráfica $G = (V, E)$ y un entero positivo $k \leq |V|$.
  Pregunta: (De hecho el complemento del conjunto independiente es *Vertex cover*).
  Existe una cubierta de vértices de tamaño k o menor para G, es decir, un subconjunto $V' \subseteq V$ tal que $|V'| \leq k$ y, que para arista $\{u, v\} \in E$, al menos una de u y v pertenezca a V'?

5. Para uno de los problemas presentados en el ejercicio 3, digamos Π, muestre que tanto Π como Π^c están en P.

   ¿Siempre sucede esto? Es decir, *¿P = CoP?* Demuestre.

   Primero se revisará el Teorema 16.1[4], que dice que si un problema A es un problema en P, entonces el complemento $\bar{A}$ de A está también en P.

   Demostración. Como A está en P, entonces hay un algoritmo polinomial que resuelve A. Un algoritmo polinomial para resolver el complemento de A es exactamente el mismo algoritmo, solo con la substitución de un *no* en vez de un *sí* cuando fue reportado, y viceversa. $\square$

6. ¿La intersección de *NP* y *coNP* es vacía? Justifica tu respuesta.

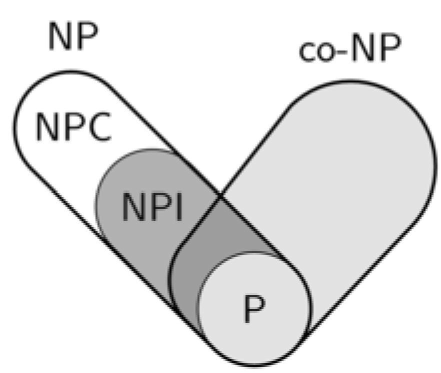


Figura 1: La intersección no es vacía

   Por que los problemas P están en la intersección de ambos.

7. Ejercicio Adicional (Puntos Adicionales) *¿NP = coNP?* ¿Quién está contenido en cuál? Justifica tu respuesta.

   Como todos los problemas en *NP* pueden ser reducidos a un problema de desición, se sigue que para cada problema en *NP* podemos construir una máquina de Turing no determinista que decide su complemento en tiempo polinomial, es decir, $NP \subseteq coNP$. De esto se sigue que que el conjunto de complemento de problemas en *coNP*, es decir, $coNP \subseteq NP$. Por lo tanto $coNP = NP$. La prueba de que ningun problema *coNP* pueda estar en *NP* si $NP \neq coNP$ e simétrica.

Bibliografía

[1] co-np - wikipedia. `https://en.wikipedia.org/wiki/Co-NP`. (Accessed on 09/17/2019).

[2] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[3] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[4] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.