Modelos Avanzados de Computación

Entrega 5

María del Mar Ruiz Martín
Doble Grado de Ingeniería Informática y Matemáticas
Universidad de Granada - UGR
18001 Granada, Spain
Curso 2016/2017

Ejercicio 6

Cada lenguaje $L \in NP \cap coNP$ sugiere un problema TFNP ¿Cuál?

Sea A el alfabeto sobre el que está definido L. Puesto que $L \in NP$ podemos afirmar que existe una relación binaria R_L de forma que podemos escribir L como sigue:

$$L = \{x \in A^* / \exists y \in A^* \ : \ xR_L y\}$$

L tiene el siguiente problema de función asociado: dado $x \in A^*$ calcular

$$\begin{cases} y \ \ tq \ \ xR_L y & si \ \{z : xR_L z\} \neq \emptyset \\ \varepsilon & en \ caso \ contrario \end{cases}$$

o equivalentemente:

$$\begin{cases} y \ \ tq \ \ xR_L y & si \ x \in L \\ \varepsilon & en \ caso \ contrario \end{cases}$$

Puesto que $L \in coNP$ sabemos que $\overline{L} \in NP$. Procediendo de forma análoga sobre $\overline{L}$ obtenemos que existe una relación binaria $R_{\overline{L}}$ tal que:

$$\overline{L} = \{x \in A^* / \exists y \in A^* \ : \ xR_{\overline{L}} y\}$$

De donde obtenemos el problema de función siguiente: dado $x \in A^*$ calcular

$$\begin{cases} y \ \ tq \ \ xR_{\overline{L}} y & si \ x \in \overline{L} \\ \varepsilon & en \ caso \ contrario \end{cases}$$

Juntando ambas expresiones conseguimos un problema TFNP: dado $x \in A^*$ calcular

$$y \ \ tq \ \ xR_L y \ \ \vee \ \ xR_{\overline{L}} y$$

Claramente ambas opciones son mutuamente excluyentes ya que $L \cap \overline{L} = \emptyset$, y ademas $L \cup \overline{L} = A^*$, por lo que el problema anterior está bien definido y efectivamente es TFNP.

Ejercicio 8

Demostrar que el siguiente problema está en P:
Dados 4 enteros a,b,c,p determinar si $a^b \equiv_p c$.

En primer lugar debemos notar que el cálculo de c módulo p es polinomial, puesto que consiste en quedarse con el resto de una división.

A continuación calcularemos a^b módulo p. Para esto usamos el mismo procedimiento visto en la demostración de que primos está en P: calculamos $a^2, a^4, a^8, ..., a^t$ módulo p, donde t es la parte entera de $log_2(b)$, que tiene orden $O(log_2(b)^3)$. Puesto que b tiene orden $O(2^n)$, siendo n la longitud de la entrada, obtenemos que esta operación tiene orden $O(log_2(2^n)^3) = O(n^3)$, y por tanto esta operación es polinómica.

Pasamos entonces al cálculo de x como el producto adecuado de las potencias anteriores. De nuevo esto será un proceso polinómico en tiempo. Dividimos el resultado obtenido por p y nos quedamos con el resto. Finalmente solo queda realizar la comprobación de si los dos números obtenidos son iguales, que es trivialmente polinómico.

Puesto que el proceso involucra tan solo procedimientos polinómicos en tiempo, podemos afirmar que el problema planteado pertenece a P.

Ejercicio 9

Sea el problema factorización que consiste en dados dos números x,y es determinar si x tiene un divisor k que sea $1 < k < y$. Demostrar que este problema está en $NP \cap coNP$.

En primer lugar veamos que el problema es NP. Consideramos una máquina de Turing que no determinísticamente genera un natural k y comprueba que $y < k$ y que k divide a x. Dichas comprobaciones son polinómicas en tiempo, y por tanto dicha máquina no determinista resuelve el problema en tiempo polinómico. Queda por tanto garantizado que el problema de la fatorización está en NP.

Para ver que también está en coNP veremos que el problema complementario es NP. Dicho problema sería el siguiente: dados x e y, determinar si x no tiene ningún divisor k tal que $1 < k < y$ o, equivalentemente, no existe ningún primo k tal que $1 < k < y$ y k divide a x. Consideramos por tanto la máquina no determinista que produce una serie de números y comprueba que todos son primos, su producto es x y que ninguno de ellos es menor que y. Todas estas comprobaciones se realizan en tiempo polinómico, puesto que se conoce que primalidad está en P, y claramente la multiplicación y la comprobación de que todos ellos son menores que y se puede realizar en tiempo polinómico. Por tanto, el problema complementario está en NP, estando así el problema factorización está en coNP.