

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2213 — Lógica para Ciencia de la Computación —
IIC2214 — Teoría de la Computación —

TAREA 6

1er semestre 2025

Pregunta 1

(a) (2.0 pts) Demuestre que si $L \leq_p K$ y $K \in \text{NP}$ entonces $L \in \text{NP}$.

Definimos coNP como la clase de todos los lenguajes L cuyo complemento $\overline{L}$ está en NP. Decimos que un problema K es coNP-hard si para todo $L \in \text{coNP}$ se tiene $L \leq_p K$. Decimos que un problema K es coNP-completo si $K \in \text{coNP}$ y K es coNP-hard.

(b) (2.0 pts) Demuestre que si un lenguaje K es NP-completo, entonces su complemento $\overline{K}$ es coNP-completo.

(c) (2.0 pts) Demuestre que si cualquier lenguaje coNP-completo está en NP, entonces $\text{NP} = \text{coNP}$.

Solución

(a) Supongamos que $L \leq_p K$ y $K \in \text{NP}$. Entonces existe una función f, computable en tiempo polinomial (digamos $O(n^k)$), tal que para toda palabra w, se cumple $w \in L \iff f(w) \in K$; y existe una MTN N que resuelve K en tiempo polinomial (digamos $O(n^\ell)$). La siguiente MTN M resuelve L:

Sobre entrada w:

1. Computar $f(w)$.
2. Simular N sobre $f(w)$.

Notar que M acepta $w \iff N$ acepta $f(w) \iff f(w) \in K \iff w \in L$.

M toma tiempo polinomial ya que el paso (1) toma tiempo $O(|w|^k)$ y el paso (2) toma tiempo $O(|f(w)|^\ell) = O(|w|^{k\ell})$. Concluimos que $L \in \text{NP}$.

(b) Sea K un lenguaje NP-completo. Veamos que $\overline{K}$ es coNP-completo. El complemento de $\overline{K}$, que es K, está en NP, luego $\overline{K} \in$ coNP. Sea $L \in$ coNP arbitrario. Tenemos que $\overline{L} \in$ NP. Como K es NP-hard, sigue que $\overline{L} \leq_p K$. Notar que si $L_1 \leq_p L_2$, entonces $\overline{L}_1 \leq_p \overline{L}_2$. (basta tomar la misma reducción). Obtenemos que $L \leq_p \overline{K}$. Como L es arbitrario, se deduce que $\overline{K}$ es coNP-hard. Concluimos que $\overline{K}$ es coNP-completo.

(c) Supongamos que existe un lenguaje coNP-completo K tal que $K \in$ NP. Veamos primero que NP $\subseteq$ coNP. Sea $L \in$ NP arbitrario. Tenemos que $\overline{L} \in$ coNP y como K es coNP-hard, se cumple que $\overline{L} \leq_p K$. Como $K \in$ NP y por la parte (a), deducimos que $\overline{L} \in$ NP. Obtenemos que $L \in$ coNP. Para ver que coNP $\subseteq$ NP, si $L \in$ coNP, entonces $\overline{L} \in$ NP, y como NP $\subseteq$ coNP, sigue que $\overline{L} \in$ coNP. Lo último implica que $L \in$ NP.

Distribución de puntajes:

(a) 1.0 pts por proponer un verificador o MTN que resuelva L correctamente, 1.0 pts por argumentar que toma tiempo polinomial.

(b) 0.5 pts por demostrar que está en coNP, y 1.5 pts por demostrar que es coNP-hard.

(c) 1.0 pts por cada contención.

Se aplican descuentos a criterio del corrector.

Pregunta 2

Decimos que una fórmula proposicional φ está en CNF_3 si está en CNF y cada variable proposicional aparece a lo más 3 veces en φ. Por ejemplo, la siguiente fórmula

$$(\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee x_3 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee x_5 \vee \neg x_6) \wedge (x_6 \vee x_7)$$

está en CNF_3 ya que cada variable aparece a lo más 3 veces, mientras que la fórmula

$$(\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee x_3 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee x_5 \vee \neg x_6) \wedge (x_6 \vee x_3)$$

no está en CNF_3 ya que x_3 aparece 4 veces.

Definamos el siguiente lenguaje:

$$\text{CNF}_3\text{-SAT} = \{\langle\varphi\rangle \mid \varphi \text{ es una fórmula proposicional satisfacible en } \text{CNF}_3\}.$$

Demuestre que CNF_3-SAT es NP-completo.

(*Hint:* Para el NP-hardness, considere hacer una reducción desde CNF-SAT. En su reducción, si una variable aparece más de 3 veces, piense en usar un grupo de variables nuevas, y modificar su fórmula de manera adecuada.)

Solución

Primero veamos que CNF_3-SAT está en NP. Basta considerar una MTN M que (1) chequee que φ está en CNF_3, (2) adivine una valuación σ para φ, (3) en caso que $\sigma(\varphi) = 1$, acepte; en caso contrario, rechace. La MTN M resuelve CNF_3-SAT. El paso (1) toma tiempo polinomial en $|\varphi|$, ya que basta revisar que cada variable aparece a lo más 3 veces en φ (por cada variable, hacemos un escaneo de la fórmula), la adivinanza del paso (2) se puede hacer en tiempo polinomial ya que el tamaño de las valuaciones σ es polinomial (a lo más $|\varphi|$) y el paso (3) también se puede hacer en tiempo polinomial, como se vio en clases.

Veamos que CNF_3-SAT es NP-hard. Demostremos que CNF-SAT $\leq_p$ CNF_3-SAT. La reducción computa para una fórmula $\varphi = C_1 \wedge \cdots \wedge C_m$ en CNF, una fórmula α_φ en CNF_3 obtenida de la siguiente forma: si una variable x ocurre ℓ veces, con $\ell \geq 4$ o más veces, entonces las ℓ ocurrencias de x se reemplazan por ℓ variables nuevas $x_1, \ldots, x_\ell$. Adicionalmente, agregamos a la fórmula, las siguientes cláusulas:

$$(x_1 \to x_2) \wedge (x_2 \to x_3) \wedge \cdots \wedge (x_{\ell-1} \to x_\ell) \wedge (x_\ell \to x_1).$$

Es decir, un ciclo de implicancias sobre $x_1, \ldots, x_\ell$. Notar que $(x \to x')$ corresponde a la cláusula $(\neg x \vee x')$. Observemos primero que α_φ está en CNF_3. Cada variable x que aparece 4 o más veces en φ se reemplaza por variables distintas $x_1, \ldots, x_\ell$, y cada x_i aparece 2 veces más, en el ciclo de implicancias. En total, cada variable en α_φ aparece a lo más 3 veces. Notemos también que la reducción se puede calcular en tiempo polinomial: para cada variable que aparece 4 o más veces, basta reemplazar sus ocurrencias por $x_1, \ldots, x_\ell$, y agregar $\ell \leq |\varphi|$ cláusulas adicionales. La cantidad de cláusulas agregadas en total es a lo más $|\varphi|^2$.

Finalmente, veamos que $\langle\varphi\rangle \in$ CNF-SAT $\iff$ $\langle\alpha_\varphi\rangle \in \text{CNF}_3$-SAT. Supongamos que $\langle\varphi\rangle \in$ CNF-SAT. Luego existe una valuación σ que satisface a φ. Definamos la siguiente valuación τ para α_φ: si x aparece a lo más 3 veces en φ, entonces sigue teniendo el mismo valor de verdad, es decir, $\tau(x) = \sigma(x)$. Si x aparece 4 o más veces en φ, entonces todas las variables $x_1, \ldots, x_\ell$ tienen el mismo valor que x, es decir, $\tau(x_1) = \tau(x_2) = \cdots = \tau(x_\ell) = \sigma(x)$. La valuación τ satisface a α_φ ya que las cláusulas originales de φ siguen teniendo el mismo valor que antes (es decir, siguen siendo verdaderas), y todos los ciclos de implicancias se hacen verdaderos ya que, o bien todas las variables son verdaderas o todas son falsas. Concluimos que $\langle\alpha_\varphi\rangle \in \text{CNF}_3$-SAT.

Para la otra dirección, supongamos que $\langle\alpha_\varphi\rangle \in \text{CNF}_3$-SAT, luego existe valuación τ que satisface a α_φ. Si una variable x aparece 4 o más veces en φ, entonces se debe cumplir que $\tau(x_1) = \cdots = \tau(x_\ell)$, debido al ciclo de implicancias que hay en α_φ. Esto nos permite definir una valuación σ para φ como sigue: si x aparece a lo más 3 veces en φ, entonces $\sigma(x) = \tau(x)$. Si x aparece 4 o más veces en φ, entonces $\sigma(x) = \tau(x_1) = \tau(x_2) = \cdots = \tau(x_\ell)$. Las valuaciones σ y τ le asignan el mismo valor de verdad a cada cláusula de φ, luego obtenemos que σ satisface a φ. Concluimos que $\langle\varphi\rangle \in$ CNF-SAT.

Distribución de puntajes:

- 1.0 pts por demostrar que $\text{CNF}_3\text{-SAT} \in \text{NP}$. 1.5 pts dar una reducción correcta. 0.5 pts por argumentar que la reducción es polinomial. 1.5 pts por cada dirección de la correctitud.

Se aplican descuentos a criterio del corrector.