Modelos de Computación y de Complejidad

Grado en Ingeniería Informática. Tecnologías Informáticas (Curso 2021–2022)

ETS Ingeniería Informática - Universidad de Sevilla

COMPLEJIDAD COMPUTACIONAL

Ejercicio 1.– Diseñar una MTD, M, cuyo alfabeto de trabajo sea $\{0, 1, B, \triangleright\}$ de tal manera que:

(a) Calcule la función identidad.

(b) Para cada dato de entrada, realice el menor número posible de transiciones (es decir, pasos de computación).

Justifíquese que la máquina diseñada satisface los requisitos exigidos.

Ejercicio 2.– Diseñar una MTD, M, cuyo alfabeto de trabajo sea $\{0, 1, B, \triangleright\}$ y que decida el lenguaje $L = \{u = \{0, 1\}^+ \mid u \text{ representa un número impar }\}$. Hállese el coste en tiempo de la máquina diseñada.

Ejercicio 3.– Diseñar una MTD, M, cuyo alfabeto de trabajo sea $\{0, 1, B, \triangleright\}$ y que decida el lenguaje $L = \{u \in \{0, 1\}^+ \mid \text{la logitud de } u \text{ es } 4 \}$. Hállese el coste en tiempo de la máquina diseñada.

Ejercicio 4.– Diseñar una MTD, M, cuyo alfabeto de trabajo sea $\{0, 1, B, \triangleright\}$ de tal manera que:

(a) Calcule la función constante e igual a 3 (de aridad 1).

(b) Para cada dato de entrada, realice el menor número posible de transiciones (es decir, pasos de computación).

Hallar el coste en tiempo de la máquina diseñada y justifíquese que la máquina diseñada satisface los requisitos exigidos.

Ejercicio 5.– Diseñar una MTD, M, cuyo alfabeto de trabajo sea $\{0, 1, B, \triangleright\}$ y que verifique la siguiente condición: existe $c \in \mathbb{N}, c > 0$ tal que para cadena $u \in \{0, 1\}^+$ de longitud n, la computación de la máquina con entrada u realice un número de pasos de computación comprendido, asintóticamente, entre $c \cdot n^2$ y $(c + 1) \cdot n^2$.

Ejercicio 6.– Diseñar una MTD, M, cuyo alfabeto de trabajo sea $\{0, 1, B, \triangleright, \#\}$ y que decida el lenguaje $L = \{(u, w) \in \{0, 1\}^+ \times \{0, 1\}^+ \mid u \text{ y } w \text{ representan números de distinta paridad} c\}$. ¿Cuál es el coste en tiempo de la máquina diseñada?

Indicación: Las cadenas de entrada de la máquina serán del tipo $u\#w$, para $(u, w) \in \{0, 1\}^+ \times \{0, 1\}^+$.

Ejercicio 7.– Diseñar una MTD, M, cuyo alfabeto de trabajo sea $\{0, 1, B, \triangleright\}$ y de tal manera que: (a) si el dato de entrada es **par**, entonces **para y devuelve 1**; y (b) si es **impar** entonces **no para**. Ilustrar el diseño realizado, considerando varios datos de entrada de tamaño 3, detallando las correspondientes computaciones.

Ejercicio 8.– Diremos que una cadena $u \in \{0, 1\}^+$ es *alternativa* si dos dígitos consecutivos, cualesquiera, siempre son distintos. Diseñar una MTD, M, cuyo alfabeto de trabajo sea $\{0, 1, B, \triangleright\}$ de tal manera que decida el lenguaje $L = \{u = \{0, 1\}^+ \mid u \text{ es una cadena } \textit{alternativa} \}$. Hállese el coste en tiempo de la máquina diseñada.

Ejercicio 9.– Sea $\{f_e \mid e \in \mathbb{N}\}$ una sucesión infinita de funciones 1-arias sobre $\mathbb{N}$ definida como sigue:

$$f_e(x) = \begin{cases} x^2 \ , & \text{si} \quad M_e(x)\downarrow \\ \uparrow \ , & \text{si} \quad M_e(x)\uparrow \end{cases}$$

Demostrar que dicha sucesión es una medida de complejidad.

Ejercicio 10.– Sean $\{f_e \mid e \in \mathbb{N}\}$ y $\{g_e \mid e \in \mathbb{N}\}$ dos medidas de complejidad. Consideremos la sucesión infinita de funciones 1-arias sobre $\mathbb{N}$, $\{h_e \mid e \in \mathbb{N}\}$, definida como sigue: $h_e(x) = f_e(x) + g_e(x)$, para cada $x \in \mathbb{N}$. Demostrar que la sucesión $\{h_e \mid e \in \mathbb{N}\}$ también es una medida de complejidad.

Ejercicio 11.– Sean $\{f_e \mid e \in \mathbb{N}\}$ y $\{g_e \mid e \in \mathbb{N}\}$ dos medidas de complejidad. Consideremos la sucesión infinita de funciones 1-arias sobre $\mathbb{N}$, $\{h_e \mid e \in \mathbb{N}\}$, definida como sigue: $h_e(x) = f_e(x) \cdot g_e(x)$, para cada $x \in \mathbb{N}$. Demostrar que la sucesión $\{h_e \mid e \in \mathbb{N}\}$ también es una medida de complejidad.

Ejercicio 12.– Sean $\{f_e \mid e \in \mathbb{N}\}$ y $\{g_e \mid e \in \mathbb{N}\}$ dos medidas de complejidad. Consideremos la sucesión infinita de funciones 1-arias sobre $\mathbb{N}$, $\{h_e \mid e \in \mathbb{N}\}$, definida como sigue: $h_e(x) = f_e((g_e(x))$, para cada $x \in \mathbb{N}$. La sucesión $\{h_e \mid e \in \mathbb{N}\}$, ¿es una medida de complejidad?

Ejercicio 13.– Un **factor de escala** es una función total computable 1-aria sobre $\mathbb{N}$ que es creciente y su rango no está acotado (la función converge a $+\infty$). Sea $\{f_e \mid e \in \mathbb{N}\}$ una medida de complejidad y g un **factor de escala** . Consideremos la sucesión infinita de funciones 1-arias sobre $\mathbb{N}$, $\{h_e \mid e \in \mathbb{N}\}$, definida como sigue: $h_e(x) = g(f_e(x))$, para cada $x \in \mathbb{N}$. Demostrar que $\{h_e \mid e \in \mathbb{N}\}$ también es una medida de complejidad.

Ejercicio 14.– EL problema **TRIÁNGULO** es el siguiente:

> *Dado un grafo no dirigido, determinar si contiene algún triángulo (subgrafo completo de tamaño 3).*

Probar que el problema **TRIÁNGULO** pertenece a la clase **P**.

Ejercicio 15.– EL problema **CONEXO** es el siguiente:

> *Dado un grafo no dirigido, determinar si es conexo.*

Probar que el problema **CONEXO** pertenece a la clase **P**.

Ejercicio 16.– Demostrar que el problema **2-SAT** pertenece a la clase **P**.

Ejercicio 17.– Explicitar una reducibilidad en tiempo polinomial del problema 2-**COL** en el problema 2-**SAT**.

Indicación: A cada grafo no dirigido se le asocia una fórmula tal que cada nodo se identifica con una variable proposicional, y cada arista $\{u, w\}$ proporciona dos cláusulas en la fórmula: $u + w$ y $\overline{u} + \overline{w}$.

Ejercicio 18.– Explicitar una reducibilidad en tiempo polinomial del problema 2-**SAT** en el problema 2-**COL**.

Ejercicio 19.– Diremos que un grafo no dirigido, $G = (V, E)$ es **bipartito** si existen $V_1, V_2 \subseteq V$ tales que $V_1 \cap V_2 = \emptyset$ y para cada $\{u, v\} \in E$ se tiene que

$$(u \in V_1 \land v \in V_2) \lor (u \in V_2 \land v \in V_1)$$

Probar que un grafo no dirigido es bipartito si y sólo si es 2–coloreable.

Ejercicio 20.– Dar ejemplos de dos problemas de decisión X_1 y X_2 tales que pertenecen a la clase **P** y, además, ninguno de ellos es reducible en tiempo polinomial, al otro.

Ejercicio 21.– Supongamos que si X_1 y X_2 son problemas de decisión tales que $X_1 \leqslant^p X_2$ y X_2 es **NP**-completo, entonces X_1 también es **NP**-completo. Demostrar que $\mathbf{P} = \mathbf{NP}$

Ejercicio 22.– Si $X = (\Sigma_X, E_X, \theta_X)$ es un problema de decisión, entonces el **problema complementario** de X es $\overline{X} = (\Sigma_X, E_X, \neg\theta_X)$. Si $\mathcal{C}$ es una clase de complejidad entonces la **clase complementaria** de $\mathcal{C}$ es **co**–$\mathcal{C} = \{X \mid \overline{X} \in \mathcal{C}\}$. Demostrar que: $\mathbf{P} = \mathbf{co\text{-}P}$ y que $\mathbf{P} \subseteq \mathbf{co\text{-}NP}$.

Ejercicio 23.– Si $\mathcal{C}$ es una clase de complejidad entonces se dice que un problema de decisión X es $\mathcal{C}$–**completo** **sii** es $\mathcal{C}$–duro (para cada $Z \in \mathcal{C}$ se tiene que $Z \leqslant^p X$) y, además, pertenece a la clase $\mathcal{C}$.

Probar que si X_1 y X_2 son problemas de decisión entonces se verifica:

(a) $X_1 \leqslant^p X_2 \;\leftrightarrow\; \overline{X_1} \leqslant^p \overline{X_2}$.

(b) X_1 es **NP**–duro si y sólo si $\overline{X_1}$ es **co-NP**–duro.

(c) X_1 es **NP**–completo si y sólo si $\overline{X_1}$ es **co-NP**–completo.

Ejercicio 24.– Demostrar que si existe un problema de decisión que es **NP**–completo y, además, pertenece a la clase **co-NP**, entonces **NP**=**co-NP**. ¿Es cierto el recíproco? Es decir, si **NP**=**co-NP** ¿podemos asegurar que existe un problema de decisión **NP**–completo que pertenezca a la clase **co-NP**?

Nota: Es una cuestión abierta si la clase **NP** coincide con la clase **co-NP**.

Ejercicio 25.– Probar que si $\mathbf{P} = \mathbf{NP}$, entonces $\mathbf{NP} = \mathbf{co\text{-}NP}$.

Ejercicio 26.– Sean $\mathcal{C}_1$ y $\mathcal{C}_2$ clases de complejidad que son **estables** bajo reducibilidad en tiempo polinomial; es decir, tales que si $X \in \mathcal{C}_i$ y $Z \leqslant^p X$ entonces $Z \in \mathcal{C}_i$. Demostrar que si existe un problema que sea, a la vez, $\mathcal{C}_1$–completo y $\mathcal{C}_2$–completo, entonces $\mathcal{C}_1 = \mathcal{C}_2$.

Ejercicio 27.– Consideremos el siguiente problema X:

> *Dada una fórmula proposicional, φ, en FNC determinar si es una tautología (o válida); es decir, si para cualquier valoración, σ, se tiene que $\sigma(\varphi) = 1$.*

Probar que el problema X es **co**–**NP**–completo.

Ejercicio 28.– Consideremos el siguiente problema X:

> *Dada una fórmula proposicional, φ, en FNC determinar si existe una valoración, σ, tal que $\sigma(\varphi) = 1$ y, además, existan variables x, y de φ tales que $\sigma(x) = 1$ y $\sigma(y) = 0$.*

Probar que el problema X es **NP**–completo.

Indicación: Pruébese que **SAT** $\leqslant^p X$.

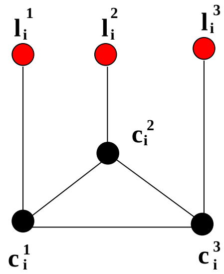
Ejercicio 29.– Consideremos el siguiente problema X:

> *Dada una fórmula proposicional, φ, en FNC determinar si existen dos valoraciones σ_1, σ_2 distintas sobre φ (es decir, para alguna variable, x, de φ se tiene que $\sigma_1(x) \neq \sigma_2(x)$) tales que $\sigma_1(\varphi) = \sigma_2(\varphi) = 1$.*

Probar que el problema X es **NP**–completo.

Ejercicio 30.– Se considera la aplicación $F : E_{\mathbf{3-SAT}} \longleftrightarrow E_{\mathbf{VC}}$ definida como sigue: Si φ entonces $F(\varphi) = (G_\varphi, n + 2p)$ (n es el número de variables y p es el número de cláusulas de φ). El grafo, $G_\varphi = (V_\varphi, E_\varphi)$ es el siguiente:

* A cada variable, x, de φ se le asocian dos nodos x y $\overline{x}$ conectados entre sí.
* Para cada cláusula $c_i = l_i^1 + l_i^2 + l_i^3$, se consideran tres nuevos nodos c_i^1, c_i^2, c_i^3, conectados entre sí y con los literales de la cláusula, como indica la figura:



Demostrar que F es una reducibilidad en tiempo polinomial de 3-**SAT** en **VC**.

Ejercicio 31.– Consideremos el siguiente problema X:

> *Dada una fórmula proposicional, φ, en FNC determinar si existen dos valoraciones σ_1, σ_2 distintas sobre φ (es decir, para alguna variable, x, de φ se tiene que $\sigma_1(x) \neq \sigma_2(x)$) tales que $\sigma_1(\varphi) = \sigma_2(\varphi) = 1$.*

Probar que el problema X es **NP**–completo.

Indicación: Pruébese que **SAT** $\leqslant^p X$.

Ejercicio 32.– Consideremos el siguiente problema X:

> *Dada una fórmula proposicional, φ, en FNC determinar si existen dos valoraciones, σ_1 y σ_2, tales que $\sigma_1(\varphi) = 1$ y $\sigma_2(\varphi) = 0$.*

Probar que el problema X es **NP**–completo.

Indicación: Pruébese que **SAT** $\leqslant^p X$.

Ejercicio 33.– Consideremos el siguiente problema X:

> *Dada una fórmula proposicional, φ, en FNC determinar si existen dos valoraciones σ_1, σ_2 distintas sobre φ, y dos variables distintas, z_1, z_2, de φ tales que $\sigma_1(\varphi) = \sigma_2(\varphi) = \sigma_1(z_1) = \sigma_1(z_2) = 1$ y $\sigma_2(z_1) = \sigma_2(z_2) = 0$.*

Probar que el problema X es **NP**–completo.

Indicación: Pruébese que **SAT** $\leqslant^p X$.

Ejercicio 34.– Dado un grafo no dirigido, $G = (V, E)$, diremos que $V_1 \subseteq V$ es un **conjunto independiente de vértices** de G sii para cada $u \in V_1, v \in V_1$ se tiene que $\{u, v\} \notin E$.

EL problema **CI** es el siguiente: *"Dado un grafo no dirigido, G, y un número natural k, determinar si existe un conjunto independiente de vértices de G de tamaño k"*.

1.- Probar que el problema **CI** pertenece a la clase **NP**.

2.- Probar que **VC** $\leqslant^p$ **CI**.

Ejercicio 35.– En la última transparencia del tema 5, se establece una reducibilidad en tiempo polinomial del problema **CLIQUE** en el problema **VC**. Basándose en la demostración que se presenta en la citada transparencia, establecer una reducibilidad en tiempo polinomial del problema **VC** en el problema **CLIQUE**.

Ejercicio 36.– Justificar la veracidad o falsedad de las siguientes afirmaciones.

1. Las computaciones de una MTD con un dato de entrada realizan, al menos, un paso de transición (o computación)

2. Si un problema abstracto es resoluble mediante una MTD con siete cintas, entonces también es resoluble mediante una MTD que posee una única cinta.

3. Las MTNDs se pueden considerar como una generalización (o extensión) de las MTDs.

4. Todas las computaciones de una MTD cuyo alfabeto de trabajo sea $\{0, 1, B, \triangleright\}$ y calcula la **función vacía**, ha de realizar, al menos, 75 pasos de transición.

5. Si M es una MTND de decisión y el coste en tiempo verifica $t_M(4) = 15$ entonces, ejecutando 16 pasos de cualquier computación de M sobre un dato de entrada de tamaño 4, sabremos **con toda seguridad** si aceptamos o no dicho dato.

6. Si M es una MTD de decisión y el coste en tiempo verifica $t_M(7) = 1115$, entonces ejecutando, a lo sumo, 111 pasos de cualquier computación de M sobre un dato de entrada de tamaño 7, sabremos **con toda seguridad** si aceptamos o no dicho dato.

7. Si una familia de funciones 1–arias sobre $\mathbb{N}$, $\{f_n : n \in \mathbb{N}\}$, es una medida de complejidad, entonces todas las funciones f_n son `GOTO`-computables.

8. La clase $\{\varphi_e^{(1)} : e \in \mathbb{N}\}$ de todas las funciones computables 1–arias sobre $\mathbb{N}$, es una medida de complejidad.

9. Si X_1 y X_2 son problemas **NP**-completos tales que $X_1 \in \mathbf{P}$ entonces $X_2 \in \mathbf{P}$.

10. Si $X_1 \in \mathbf{P}$ y X_2 es un problema **NP**-completo que es reducible en tiempo polinomial a X_1 entonces $X_2 \in \mathbf{P}$.

11. Si X_1 y X_2 son problemas de la clase **P**, entonces se tiene que $X_1 \leqslant^p X_2$ o bien quer $X_2 \leqslant^p X_1$.

12. Si X_1 y X_2 son problemas **NP**-completos entonces **cualquiera de ellos** es *reducible en tiempo polinomial* al otro.

13. Si X_1, X_2 son problemas de decisión tales que X_1 es *reducible en tiempo polinomial* al problema X_2 y, además, X_2 pertenece a la clase **NP**, entonces el problema X_1 también pertenece a la clase **NP**.

14. Si un problema X pertenece a la clase **P** entonces *todo problema más difícil* que X **también** pertenece a la clase **P**.

15. Sean X_1 y X_2 problemas de decisión tales que X_1 es un problema **NP**-completo y $X_1 \leqslant^p X_2$. Entonces, X_2 también es **NP**-completo.

16. Si X_1 y X_2 son problemas de la clase **P** y, además, X_1 es **NP**-completo entonces X_2 también es **NP**-completo.

17. Los problemas **NP**-duros tienen la propiedad de ser más difíciles que cualquier otro problema perteneciente a la clase **NP**.

18. Los problemas **NP**-completos son idóneos para atacar el problema **P** versus **NP**.

19. El problema **CLIQUE** pertenece a la clase **NP**.

20. La resolución mecánica del problema **CLIQUE** y del problema **VC** (del recubrimiento de vértices) tienen el mismo grado de dificultad comparativa.