# MODELOS AVANZADOS DE COMPUTACIÓN

**Relación 1**

1. Diseñar máquinas de Turing para los siguientes lenguajes:

   (a) Palabras sobre el alfabeto $\{0, 1\}$ con el mismo número de ceros que de unos.

   (b) $L = \{a^n b^n c^n | n \geq 1\}$

   (c) $\{ww^{-1} \mid w \in \{0, 1\}^*\}$

   (d) $\{wcw \mid w \in \{0, 1\}^*\}$

2. Rediseñar las MTs del ejercicio anterior introduciendo las técnicas de programación de almacenamiento de símbolo, multipista o subrutinas.

3. Diseñar una subrutina que desplace todos los símbolos desde la posición actual a la derecha, dejando un espacio en dicha posición en el que se pueda escribir un carácter. Debe de terminar con la cabeza de lectura en la misma posición en la que se empezó.

4. Escribir una subrutina que comience en una posición con un cero y se mueva a la derecha de todos los ceros hasta que alcance un uno o un blanco. Se suponen que en la cinta solo hay caracteres del conjunto $\{0, 1, \#\}$. Si se comienza con un carácter que es distinto de cero, la subrutina de la MT debe de pararse. Utilizar dicha rutina para escribir una MT que acepte todas las cadenas de ceros y unos, que no tengan dos unos consecutivos.

5. Diseñar una MT con dos cintas que dada una suceción de de ceros de longitud $n$ en la primera cinta, calcula en la segunda cinta $n$ en binario.

6. Escribir una MT con múltiples cintas que sume dos números en binario. Se supone que aparecen en la cinta de entrada separados por un símbolo especial $c$.

7. Describir una MT que dada una palabra $ucw$ donde $u$ y $w$ son dos palabras sobre el alfabeto $\{0, 1\}$ y $c$ es un símbolo adicional, calcule $u$ repetido tantas veces como indique la palabra $w$ interpretándola como un entero escrito en binario.

8. Supongamos la MTND $M = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, \#\}, \delta, q_0, \#, \{q_2\})$ donde

| $\delta$ | 0 | 1 | # |
|---|---|---|---|
| $q_0$ | $\{(q_0, 1, D)\}$ | $\{(q_1, 0, D)\}$ | $\emptyset$ |
| $q_1$ | $\{(q_1, 0, D), (q_0, 0, I)\}$ | $\{(q_1, 1, D), (q_0, 1, I)\}$ | $(q_2, \#, D)$ |
| $q_2$ | $\emptyset$ | $\emptyset$ | $\emptyset$ |

   Estudiar las configuraciones que se pueden alcanzar si la palabra de entrada es:

(a) 01

(b) 011

9. Describir MTND (con una o varias cintas) que acepten los siguientes lenguajes:

   (a) Conjunto de palabras que contienen una subcadena de longitud 100 que se repite aunque no necesariamente de forma consecutiva.

   (b) El conjunto de las cadenas $w_1 \circ w_2 \circ \cdots \circ w_n$ donde $w_i \in \{0,1\}^*$ y para algún $j$ $w_j$ coincide con la representación en binario de $j$.

   (c) El conjunto de las cadenas $w_1 \circ w_2 \circ \cdots \circ w_n$ donde $w_i \in \{0,1\}^*$ y para, al menos, dos valores de $j$ $w_j$ coincide con la representación en binario de $j$.

10. Se considera la MTND $M = (\{q_0, q_1, q_2, q_f\}, \{0,1\}, \{0,1,\#\}, \delta, q_0, \#, \{q_f\})$ con

$$\begin{array}{ll} \delta(q_0,0) = \{(q_0,1,D),(q_1,1,D)\} & \delta(q_1,1) = \{(q_2,0,I)\} \\ \delta(q_2,1) = \{(q_0,1,D)\} & \delta(q_1,\#) = \{(q_f,\#,D)\} \end{array}$$

Describir el lenguaje generado.

# MODELOS AVANZADOS DE COMPUTACIÓN

**Relación 2**

1. Sobre el alfabeto $A = \{a, b, c\}$, calcular las palabras $C(143)$ y $C(100)$.

2. Sobre el alfabeto $A = \{a, b, c\}$, calcular las palabras $Z(aabc)$ y $C(bac)$.

3. Discutir la posibilidad de asignar un número natural a cada MT con independencia del alfabeto de entrada.

4. Construir una MT que dada una entrada $w$, la convierte en una salida $w111w$.

5. Demostrar que el problema de la parada: determinar el conjunto de parejas $(M, w)$ tales que la MT $M$ para cuando tiene a $w$ como entrada es r.e. pero no recursivo.

6. Describir de manera informal MTs con varias cintas que enumeren (produzcan como salida una lista que contenga todas sus palabras) los siguientes lenguajes:

   (a) El conjunto de los cuadrados perfectos.

   (b) El conjunto de todos los naturales primos.

   (c) El conjunto de todos los números naturales $n$ tales que $M(n)$ acepta la palabra $C(n)$.

7. Sean $L1, \ldots, L_k$ un conjunto de lenguajes sobre el alfabeto $A$ tales que:

   (a) Para cada $i \neq j$, tenenos que $L_i \cup L_j = \emptyset$.

   (b) $\bigcup_{i=1}^{k} L_i = A^*$.

   (c) $\forall i \in \{1, \ldots, k\}$, el lenguaje $L_i$ es r.e.

   Demostrar que $\forall i \in \{1, \ldots, k\}$, el lenguaje $L_i$ es recursivo.

8. Sea $L$ r.e., pero no recursivo. Considérese el lenguaje

$$L' = \{0w \mid w \in L\} \cup \{1w \mid w \notin L\}$$

   ¿Puede asegurarse que $L$ o su complementario son recursivos, r.e. o no r.e.?

9. Estudiar si las clases de lenguajes recursivos y r.e. son cerradas para las siguientes operaciones:

   (a) Unión.

   (b) Intersección.

   (c) Concatenación.

(d) Clausura.

(e) Homomorfismo.

(f) Homomorfismo inverso.

10. Demostrar que el conjunto de las MT que aceptan al menos un palíndromo es indecidible.

11. Supongamos que tenemos MT con dos tipos de estados: *campana* y *silbato*. Determinar que saber si una MT entrará alguna vez en un estado *silbato* es indecidible.

12. Demostrar que es indecidible (no recursivo) saber si una MT termina escribiendo un 1 cuando comienza con una cinta completamente en blanco.

13. Determinar si los siguientes lenguajes son recursivos, r.e. o no r.e.:

    (a) Determinar si el lenguaje de una MT contiene, al menos, dos palabras distintas.

    (b) Determinar si el lenguaje de una MT es finito o infinito.

    (c) Determinar si el lenguaje de una MT es independiente del contexto.

14. Sea $L$ el lenguaje formado por pares de códigos de MT más un entero $(M_1, M_2, k)$ tales que $L(M_1) \cap L(M_2)$ contiene, al menos, $k$ palabras. Demostrar que $L$ es r.e. pero no recursivo.

15. Demostrar que las siguientes cuestiones son decidibles:

    (a) El conjunto de las MT $M$ tales que al comenzar con la cinta en blanco, en algún momento escribirán un símbolo no blanco en la cinta.

    (b) El conjunto de las MT que nunca se mueven a la izquierda.

    (c) El conjunto de los pares $(M, w)$ tales que $M$, al actuar sobre la entrada $w$, nunca lee una casilla de la cinta más de una vez.

16. Indicar si los siguientes lenguajes son recursivos, r.e. o no r.e.

    (a) El conjunto de los códigos de MT que se paran para cualquier entrada.

    (b) El conjunto de los códigos de MT que no se paran para ninguna entrada.

    (c) El conjunto de los códigos de MT que se paran, al menos, para una entrada.

    (d) El conjunto de los códigos de MT que no se paran, al menos, para una entrada.

# MODELOS AVANZADOS DE COMPUTACIÓN

**Relación 3**

1. Construir un programa Post-Turing que calcule la función $f(u) = u^{-1}$ donde $u \in \{0,1\}^*$.

2. Construir un programa Post-Turing que dado un número $u$ en binario calcule $u + 1$.

3. Construir un programa Post-Turing que dadas dos cadenas $ucv$ donde $u, v \in \{0,1\}^*$ calcule si la cadena $u$ es una subcadena de la cadena $v$.

4. Construir un programa con variables que concatene dos cadenas sobre $\{0,1\}$. Se supone que ambas cadenas están en las variables $X_1$ y $X_2$ y la salida en la variable $Y$.

5. Construir un programa con variables que dadas dos cadenas sobre $\{0,1\}$ (en las variables $X_1$ y $X_2$) calcule el número de apariciones de $X_1$ como subcadena de $X_2$. La salida será un número en binario en $Y$.

6. Construir un programa con variables que acepte el lenguaje $L = \{w \in \{0,1\}^* \mid w = w^{-1}\}$.

7. Construir un programa con cadenas que dada una cadena $u \in \{0,1\}^*$ calcule la cadena $w$ formada por los símbolos que ocupan las posiciones impares de $u$ y en el mismo orden que aparecen en $u$.

8. Construir un programa con cadenas sobre $\{0,1\}$ que dadas dos cadenas $u_1, u_2 \in \{a,b\}^*$ calcule la cadena $u$ cuyo número $Z(u) = Z(u_1) + Z(u_2)$.

9. Considerar un lenguage Post Turing para programas con varias cintas. Hay un número finito de cintas y en cada momento una de ellas está activa, inicialmente la primera. Hay dos instrucciones UP and DOWN que se mueven a la cinta superior e inferior respectivamente. Demostrar que todo cálculo realizado por un programa Post Turing con varias cintas, puede realizarse con un programa Post Turing con una sola cinta.

10. Dado el siguiente programa con variables:

```
     IF X ENDS 0 GOTO A
     IF X ENDS 1 GOTO B
     HALT
[A]  X ← X-
     Y ← 0Y
     IF X ENDS 0 GOTO A
     IF X ENDS 1 GOTO B
```

```
        HALT
[B]     X ← X-
        Y ← 1Y
        IF X ENDS 0 GOTO A
        IF X ENDS 1 GOTO B
        HALT
```

construir un programa Post-Turing equivalente (se pueden usar macros).

11. Dado el siguiene programa Post-Turing

```
        LEFT
[C]     RIGHT
        IF # GOTO E
        IF 0 GOTO A
        IF 1 GOTO C
[A]     PRINT #
        IF # GOTO C
[E]     HALT
```

construir una MT equivalente.

12. Dada la MT $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, \#\}, \delta, q_0, \#, \{q_4\})$ donde las transiciones no nulas son las siguientes:

$$\begin{array}{ll}
\delta(q_0, 0) = (q_1, X, D) & \delta(q_0, Y) = (q_3, Y, D) \\
\delta(q_1, 0) = (q_1, 0, D) & \delta(q_1, 1) = (q_2, Y, I) \\
\delta(q_1, Y) = (q_1, Y, D) & \delta(q_2, 0) = (q_2, 0, I) \\
\delta(q_2, X) = (q_0, X, D) & \delta(q_2, Y) = (q_2, Y, I) \\
\delta(q_3, Y) = (q_3, Y, D) & \delta(q_3, \#) = (q_4, \#, D)
\end{array}$$

construir un programa con variables equivalente (se pueden usar macros).

# MODELOS AVANZADOS DE COMPUTACIÓN

**Relación 4**

1. Construir un programa con variables numéricas que calcule $f(x_1, x_2) = x_1 + x_2$ y otro que calcule $f(x_1, x_2) = x_1 x_2$.

2. Escribir un programa con variables numéricas que calcule $f(x) = 1$ si $x$ es par y 0 en caso contrario.

3. Escribir un programa con variables numéricas que $f(x) = 1$ si $x$ es primo y 0 en caso contrario.

4. Escribir un programa con variables numéricas que calcule $f(x) = y$ donde $y = Z(C(x)-)$ donde $Z$ y $C$ son las codificaciones sobre un alfabeto de $n$ símbolos.

5. Escribir un programa con variables numéricas que calcule $f(x) = y$ donde $y = Z(a_i C(x))$ donde $Z$ y $C$ son las codificaciones sobre un alfabeto de $n$ símbolos.

6. Demostrar que las siguientes funciones son primitivo recursivas:

   (a) Exponenciación: $Exp(x, y) = x^y$

   (b) División entera: $Div(x, y) = [x/y]$

   (c) Resto de la división entera: $R(x, y)$

   (d) Función predecesor: $p(x) = x - 1$ (si $x > 1$) y $p(0) = 0$.

   (e) Resta $Resta(x, y) = x \dot{-} y$ (la resta, resultando 0 si $x \leq y$).

   (f) Comparación $Compara(x, y)$ con resultado 1 si $x > y$ y 0 en caso contrario.

# MODELOS AVANZADOS DE COMPUTACIÓN

**Relación 5**

**Clases de Complejidad**

1. (a) Un grafo dirigido se dice que es *acíclico* si no tiene ciclos. Demostrar que todo grafo dirigido acícilo tiene una *fuente* (un nodo al que no llegan arcos).

   (b) Demostrar que un grafo dirigido con $n$ nodos es acíclico si y solo si se pueden numerar los nodos del 1 al $n$ de manera que siempre los arcos van desde números más pequeños a números más grandes.

   (c) Describir un algoritmo polinómico para determinar cuando un grafo es acíclico.

2. (a) Demostrar que un grafo es *bipartito* (se puede dividir en dos partes, que no son necesariamente iguales, de manera que todos las aristas van de una parte a otra) si y solo si todos sus ciclos son de longitud par.

   (b) Describir un algoritmo polinómico para comprobar si un grafo es bipartito.

3. Demostrar que **P** es cerrada para la unión y la intersección.

4. Demostrar que $\mathbf{NP} \neq \mathbf{ESPACIO}(n)$

5. Sea $C$ una clase de funciones de los enteros no negativos en los enteros no negativos. Se dice que $C$ es cerrada para la composición polinómica por la izquierda si $f(n) \in C$ implica que $p(f(n)) = O(g(n))$ para alguna función $g(n) \in C$, y donde $p(n)$ es un polinomio cualquiera. Se dice que $C$ es cerrada para la composición polinómica por la derecha si $f(n) \in C$ implica que $f(p(n)) = O(g(n))$ para alguna función $g(n) \in C$, y donde $p(n)$ es un polinomio cualquiera.

   Determinar cuales de las siguientes clases de funciones son cerradas para la composición polínómica por la derecha y cuales lo son por la izquierda.

   (a) $\{n^k : k > 0\}$

   (b) $\{n.k : k > 0\}$

   (c) $\{k^n : k > 0\}$

   (d) $\{2^{n^k} : k > 0\}$

   (e) $\{\log^k(n) : k > 0\}$

   (f) $\{\log(n)\}$

# MODELOS AVANZADOS DE COMPUTACIÓN

**Relación 6**

**NP-Completitud**

1. Demostrar que los siguientes problemas son NP-completos:

   *a*) **Camino más largo** Dado un grafo $G = (V, E)$ y un entero positivo $K \leq |V|$, ¿contiene $G$ un camino simple (que no pase dos veces por el mismo sitio) con $K$ o más arcos?

   *b*) **Empaquetado de Conjuntos** Dada una colección $C$ de conjuntos finitos y un entero $K \leq |C|$, ¿existen $K$ conjuntos disjuntos en $C$?

   *c*) **Partición en subgrafos hamiltonianos** Dado un grafo $G = (V, E)$, y un entero positivo $K \leq |V|$, ¿pueden particionarse los vértices de $G$ en $k \leq K$ conjuntos disjuntos $V_1, V_2, \ldots, V_k$ de tal forma que para todo $1 \leq i \leq k$, $V_i$ contiene un circuito hamiltoniano?

   *d*) **Subgrafo común maximal** Dados los grafos $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$, y un entero positivo $K$, ¿existen subconjuntos $E_1' \subseteq E_1$ y $E_2' \subseteq E_2$ tales que $|E_1'| = |E_2'| \geq K$ y tal que los dos subgrafos $G_1' = (V_1, E_1')$ y $G_2' = (V_2, E_2')$ son isomorfos?

   *e*) **Suma de cuadrados mínima** Dado un conjunto finito $A$ y un tamaño $s(a) > 0$ para todo $a \in A$ y dos enteros positivos $K$ y $J$, ¿pueden particionarse los elementos de $A$ en $K$ conjuntos disjuntos, $A_1, \ldots, A_k$, de tal forma que $\sum_{i=1}^{K} \left( \sum_{a \in A_i} s(a) \right)^2 \leq J$?

   *f*) **Conjunto de vértices de corte** Dado un grafo dirigido $G = (V, A)$, y un entero positivo $K \leq |V|$, ¿existe un subconjunto $V' \subseteq V$ tal que $|V'| \leq K$ y tal que todo circuito dirigido en $G$ incluya al menos un vértice de $V'$?

   *g*) **Cubrimiento exacto por conjuntos de 4 elementos** Dado un conjunto finito $X$ con $|X| = 4q$, siendo $q$ un entero y una familia $C$ de subcojuntos de 4 elementos de $X$, ¿existe una subfamilia $C' \subseteq C$ tal que todo elemento de $X$ pertenece a uno y solo uno de los subconjuntos de $C$?

   *h*) **Conjunto dominante** Dado un grafo $G = (V, E)$ y un entero positivo $K \leq |V|$, ¿existe un subconjunto $V' \subseteq V$ tal que $|V'| \leq K$ y tal que todo vértice $v \in V - V'$ está conectado con al menos un vértice de $V'$?

   *i*) **Estrella de Steiner en grafos** Dado un grafo $G = (V, E)$, y un subconjunto $R \subseteq V$, y un entero positivo $K \leq |V| - 1$, ¿existe un subárbol de $G$ que contiene todos los vértices de $R$ y que no contiene más de $K$ arcos?

   *j*) **Equivalencia de expresiones regulares sin estrella** Dadas dos expresiones regulares $E_1$ y $E_2$ sobre el alfabeto $A$ que no contienen el operador de clausura $*$, ¿representan estas expresiones regulares lenguajes distintos sobre $A$?

*k*) **Partición de conjuntos** Dada una familia $C$ de subconjuntos sobre un conjunto finito $S$ ¿existe una partición de $S$ en dos partes $S_1$ y $S_2$ tales que no hay un subconjunto de $S$ contenido en $S_1$ o $S_2$

*Nota.*- Reducir 3-SAT.

*l*) **Partición en caminos de longitud 2** Dado un grafo $G = (V, E)$, con $|V| = 3q$ donde $q$ es un entero positivo, ¿existe una partición de $V$ en $q$ conjuntos disjuntos $V_1, V_2, \ldots, V_q$ de tal forma que para cada $V_i = \{v_{i[1]}, v_{i[2]}, v_{i[3]}\}$, al menos dos de los tres posibles arcos, $\{v_{i[1]}, v_{i[2]}\}, \{v_{i[1]}, v_{i[3]}\}, \{v_{i[2]}, v_{i[3]}\}$ está en $E$.

*Nota.*- Reducir cubrimiento por tripletas

*m*) **Numeración en Grafos** Dado un grafo dirigido $G = (V, A)$, ¿existe una numeración $L : V \to \mathbb{N}$, donde el mismo número puede asignarse a más de un vértice y tal que cada $L(u)$ es igual al mínimo de todos los valores enteros que no están en $\{L(v) : (u, v) \in A\}$.

*n*) **Colorear grafos(3)** Dado un grafo $G = (V, E)$, ¿existe una función $f : V \to \{1, 2, 3\}$, tal que $f(u) \neq f(v)$ para todos los arcos $\{u, v\} \in E$?

2. Una Máquina de Turing no-determinística fuerte es una máquina que tiene tres posibles respuestas 'Si', 'No', y 'Duda'. Una de estas máquinas decide $L$ si y solo si, para para todo $x \in L$, todos los cálculos posibles terminan en 'Si' o 'Duda' y al menos uno en 'Si' y para todo $x \notin L$, todos los cálculos posibles terminan en 'No' o 'Duda' y al menos uno en 'No'. Demostrar que $L$ es decidido por una Máquina de Turing no-determinística fuerte si y solo si $L$ está en $\mathbf{NP} \cap \mathbf{coNP}$.

3. Demostrar que es NP-completo el siguiente problema: dado un grafo y un circuito hamiltoniano determinar si el grafo tiene otro circuito hamiltoniano.

4. Cada lenguaje $L \in \mathbf{NP} \cap \mathbf{coNP}$ sugiere un lenguaje en **TFNP** ¿Cual?

5. Demostrar que FSAT es **FNP**-completo.

6. Demostrar que el siguiente problema está en **P**:

   Dados 4 enteros $a, b, c, p$ determinar si $a^b \equiv c(\mathrm{mod} p)$

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