**FUNDAMENTOS DE TEORÍA DE LA COMPUTACIÓN**
**Computabilidad y Complejidad Computacional. Examen 1 Año 2023. Licenciatura en Sistemas.**
**Plantel Docente: R. Rosenfeld, L. Mendoza, P. Dal Bianco.**

---

Comentarios

El examen dura 2 horas y es a libro cerrado. No desarrollar nada ya hecho en clase, pero sí indicar de dónde proviene. Ser breves y claros. Aprovechar bien el tiempo, no dedicar mucho tiempo a ningún ejercicio en particular. Hay que hacer ejercicios de Computabilidad y de Complejidad Computacional. ¡Mucha suerte!

## COMPUTABILIDAD

**Ejercicio 1.** Construir formalmente una MT que acepte el lenguaje de las cadenas de la forma $(123)^n$, con $n \geq 0$, es decir las cadenas: λ (vacía), 123, 123123, 123123123, etc.

**Ejercicio 2.** Sean $L_1, L_2, \ldots, L_n$, lenguajes de RE disjuntos (es decir que no comparten cadenas), y cuya unión es todo $\Sigma^*$. Explicar por qué cada uno de ellos es recursivo. Ayuda: basta con explicar cómo construir una MT $M_i$, con $1 \leq i \leq n$, que acepte el lenguaje $L_i$ y pare siempre, sabiendo por hipótesis que existen MT, que pueden loopear en algunos casos, que aceptan los lenguajes $L_i$ con las características mencionadas.

**Ejercicio 3.** ¿Cuándo una función es una reducción de un lenguaje $L_1$ a un lenguaje $L_2$?

**Ejercicio 4.** Se cumple que todos los lenguajes de RE se reducen al lenguaje $L_U$. Considerando esta hipótesis, probar por qué si $L_U$ fuera recursivo, entonces se cumpliría R = RE. Ayuda: tener en cuenta una de las propiedades que estudiamos de las reducciones de lenguajes.

**Ejercicio 5.** Construir una reducción f del lenguaje de códigos de MT al conjunto de los números naturales N = {0, 1, 2, ...}, tal que f sea inyectiva (es decir que a todo código de MT le asigne un número natural distinto). Ayuda: recordar que las MT se pueden ordenar.

## COMPLEJIDAD COMPUTACIONAL

**Ejercicio 6.** Explicar cuándo un lenguaje pertenece a:
(a) La clase P.
(b) La clase NP.
(c) La clase NPC.

**Ejercicio 7.** Un conjunto independiente de vértices de un grafo G es un conjunto de vértices $\{v_1, v_2, \ldots, v_K\}$ de G no adyacentes dos a dos, es decir que para todo $v_i$ y $v_j$ del conjunto, con $i \neq j$, se cumple que $v_i$ y $v_j$ no están conectados por un arco. Sea el lenguaje CI = {(G, K) | G tiene un conjunto independiente de vértices de tamaño K}. Probar que CI pertenece a la clase NP. Ayuda: la solución en un sentido es similar a la que vimos para probar que CLIQUE está en NP.

**Ejercicio 8.** ¿Por qué el lenguaje CI del ejercicio anterior no estaría en P?

**Ejercicio 9.** Si se probara que existe una reducción polinomial de SAT a CI, ¿se estaría probando que CI es un problema NP-completo? Justificar la respuesta.

**Ejercicio 10.** En el marco de la complejidad espacial, explicar:
(a) ¿Por qué se trabaja con MT con cintas de input de solo lectura?
(b) ¿Por qué se considera que un problema es tratable si pertenece a la clase LOGSPACE?
(c) ¿Por qué P está incluido en PSPACE?

# Fundamentos de teoría de la computación 2023. Lógica. Examen 1. 04/07/2023. T2

**Ejercicio 1.** Dada la siguiente tabla de verdad:

| p | q | ? |
|---|---|---|
| V | V | F |
| V | F | F |
| F | V | F |
| F | F | F |

a) Obtener la fórmula asociada utilizando **sólo los conectivos de negación y conjunción (¬, ∧)**.
b) Obtener una fórmula **lógicamente equivalente** a la fbf obtenida en el punto a) utilizando **sólo los conectivos de negación e implicación (¬, →)**.

**Ejercicio 2.** Dada la siguiente argumentación dar una forma argumentativa que se corresponda con ella y determinar si es válida o inválida:

*"Si Gomez ha instalado la actualización, entonces el software funciona bien o tiene un error en el código. Por lo tanto, si el software no funciona bien entonces Gomez no ha instalado la actualización"*

**Ejercicio 3.** Dada la siguiente secuencia de fbfs de L:

1. $(t \to u)$ H
2. $(t \to (u \to t)) \to ((t \to u) \to (t \to t))$ $L_2$
3. $(t \to (u \to t))$ $L_1$
4. $((t \to u) \to (t \to t))$ MP
5. $(t \to t)$ A

a) Indicar si es una demostración sintáctica válida en L para algún conjunto Γ y una fórmula A tal que $\Gamma \vdash_L A$. En caso afirmativo indicar Γ, A y que axioma, hipótesis o regla de inferencia fue aplicado en cada paso de la demostración. En caso negativo, identificar los errores.
b) Es A un teorema de L? Justificar.

**Ejercicio 4.** Sea B una fbf cualquiera. ¿Es posible que se cumpla $\vdash_L B$ y al mismo tiempo B sea una contradicción? Justificar.

**Ejercicio 5.** Dado el predicado M(x,y) que representa *"x es mayor que y"* y la función *f(x) que representa "-1"*, formular en lenguaje simbólico de primer orden cada una de las siguientes expresiones:

a) Existe un número que es mayor a todos los demás.
b) Todo número es mayor que su antecesor.
c) Para todo par de números existe uno que está entre ambos.

* *Tener en cuenta que estamos trabajando sobre el dominio de los **números reales** (no es necesario especificarlo). Se pueden utilizar los cuantificadores "∀" y "∃" y conectivos lógicos si es necesario pero **NO** se deben utilizar predicados ni funciones adicionales.*

**Ejercicio 6.** Dado el predicado binario "P", la constante "a", la función "f(x)" y la fórmula A = "∀x P(f(x),a)" indicar si A es **satisfactible** en una I (Interpretación), **verdadera** en una I, **falsa** en una I, **lógicamente válida** o **contradictoria**. Justificar en cada caso.

**Ejercicio 7.** ¿Es posible escribir una fórmula **abierta** y **contradictoria**? En caso afirmativo justificar y ofrecer una fórmula que sirva como ejemplo. En caso negativo, justificar.

**Ejercicio 8.** Dados los predicados unarios P y Q, analizar si la siguiente fbf es lógicamente válida. Justificar.

$(\exists x\, P(x) \wedge \exists x\, Q(x)) \to (\exists x\, (P(x) \wedge Q(x)))$