FUNDAMENTOS DE TEORÍA DE LA COMPUTACIÓN.
Examen Nro 1 Año 2018. Licenciatura en Sistemas.
Plantel Docente: R. Rosenfeld, L. Mendoza, I. Rosenfeld

Ejercicio 1. ¿Qué postula la Tesis de Church-Turing?

Ejercicio 2. El Problema de la Partición consiste en determinar, dado un conjunto de números naturales $K = \{x_1, x_2, \ldots, x_n\}$, si existe una partición de K en dos subconjuntos K_1 y K_2 tal que la suma de los números de K_1 es igual a la suma de los números de K_2. Especificar el lenguaje que representa el problema.

Ejercicio 3. Se prueba que el modelo de máquinas de Turing (o MT) con cintas semi-infinitas (finitas a izquierda e infinitas a derecha) es equivalente al modelo de MT tratado en clase. ¿Qué significa que ambos modelos de MT son equivalentes?

Ejercicio 4. Construir una MT que acepta el lenguaje de las cadenas de 1 y 0 que empiezan con la subcadena 10. *Comentario: Hay que definir Q, Σ, Γ, δ y q_0.*

Ejercicio 5. Responder:
a) ¿Por qué $R \subseteq RE$?
b) ¿Por qué $RE \subseteq \mathfrak{L}$?
c) ¿Por qué $R \subseteq$ CO-RE?
d) Dar un ejemplo de lenguaje en RE – R y un ejemplo de lenguaje en CO-RE – R.

Ejercicio 6. Sean dos lenguajes L_1 y L_2 de RE, aceptados por MT M_1 y M_2, respectivamente, y sea la siguiente MT M, que dado un input w, hace:
1. Ejecuta M_1, y si M_1 rechaza (parando o no), entonces rechaza (parando o no).
2. Ejecuta M_2, y responde como M_2.

Se cumple que la MT M acepta la intersección entre L_1 y L_2. Justificar.

Ejercicio 7. Sea el lenguaje $D = \{w_i \mid$ la MT M_i para desde $w_i\}$, considerando el orden canónico. Probar que $D \in RE$. *Ayuda: Construir una MT que acepta el lenguaje D.*

Ejercicio 8. Sea f una función de reducción de un lenguaje L_1 a un lenguaje L_2. Responder:
a) ¿Qué significa que f es una función de reducción de L_1 a un lenguaje L_2?
b) ¿A qué clase pertenece L_1 si $L_2 \in R$? Justificar.
c) ¿A qué clase no pertenece L_2 si $L_1 \notin RE$? Justificar.

Ejercicio 9. ¿Qué diferencias existen entre las MT generales, los autómatas finitos y los autómatas con pila, en los siguientes aspectos?
a) Movimientos del cabezal sobre la cinta de input.
b) Escritura sobre la cinta de input y sobre las otras cintas.
c) Cuándo paran.
d) Cuándo aceptan y cuándo rechazan.

Ejercicio 10. Sea la clase temporal TIME(T(n)). ¿Qué significa que un lenguaje L pertenece a TIME(T(n))?

Ejercicio 11. ¿Qué postula la Tesis Fuerte de Church-Turing?

Ejercicio 12. Ya definimos antes el Problema de la Partición: consiste en determinar, dado un conjunto de números naturales $K = \{x_1, x_2, \ldots, x_n\}$, si existe una partición de K en dos subconjuntos K_1 y K_2 tal que la suma de los números de K_1 es igual a la suma de los números de K_2. Probar que el problema está en NP. *Comentario: la suma de dos números tarda tiempo poly(n).*

Ejercicio 13. Probar que si se cumple $L_1 \alpha_P L_2$, $L_2 \alpha_P L_1$ y $L_1 \in NPC$, entonces $L_2 \in NPC$. *Ayuda: usar las definiciones de reducción polinomial y de problema NP-completo.*

FUNDAMENTOS DE TEORÍA DE LA COMPUTACIÓN.
Examen Nro 1 Año 2018. Licenciatura en Sistemas.
Plantel Docente: R. Rosenfeld, L. Mendoza, I. Rosenfeld

Ejercicio 14. Se indicó en clase que si existe un lenguaje NP-completo en CO-NP, entonces NP = CO-NP, y se probó sólo la inclusión NP $\subseteq$ CO-NP. La prueba de la inclusión inversa, es decir CO-NP $\subseteq$ NP, podría hacerse de la siguiente manera:

i. Existe por hipótesis un lenguaje L_1 NP-completo que está en CO-NP.
ii. Sea L_2 algún lenguaje de CO-NP. Hay que probar que L_2 está en NP.
iii. Se cumple que $L_2^C \in$ NP.
iv. Entonces $L_2^C \alpha_P L_1$.
v. Entonces $L_2 \alpha_P L_1^C$.
vi. Entonces $L_2 \in$ NP.

Explicar cómo se obtienen los pasos iii a vi.

Ejercicio 15. Explicar por qué si una MT M_1 trabaja en espacio SPACE(S(n)), entonces existe una MT M_2 equivalente que trabaja en el mismo espacio y para siempre.

FUNDAMENTOS DE TEORÍA DE LA COMPUTACIÓN
Computabilidad y Complejidad Computacional. Examen 1 Año 2022. Licenciatura en Sistemas.
Plantel Docente: R. Rosenfeld, L. Mendoza, P. Dal Bianco.

Comentarios

- *El examen dura 2 horas y es a libro abierto.*
- *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. Lenguajes como HP, L_u, $L_\emptyset$, $L_{\Sigma*}$, etc., no son recursivos.
- a) ¿Cuándo un lenguaje no es recursivo?
- b) ¿Puede ser recursivo el complemento de un lenguaje no recursivo? Justificar.

Ejercicio 2. Construir una MT que acepte el lenguaje de las cadenas con 1 y 0 tales que tengan una cantidad impar de 1.

Ejercicio 3. ¿Qué es una MT universal? Además, describa informal y brevemente cómo trabaja, a partir de un input (<M>, w).

Ejercicio 4. Probar que si $L_1 \in RE$ y $L_2 \in R$, entonces $L_1 - L_2 \in RE$.
Ayuda: construir una MT que acepte $L_1 - L_2$, o sea las cadenas que están en L_1 y no están en L_2.

Ejercicio 5. La siguiente función f es una reducción del lenguaje $L_{\Sigma^*} = \{<M> \mid L(M) = \Sigma^*\}$ al lenguaje $L_{EQ} = \{(<M_1>, <M_2>) \mid L(M_1) = L(M_2)\}$:

$$f(<M>) = (<M>, <M_{\Sigma^*}>), \text{ siendo } M_{\Sigma^*} \text{ una MT que acepta } \Sigma^*$$

Comprobar que efectivamente f es una reducción. En otras palabras, probar:
- a) f es una función total computable.
- b) $<M> \in L_{\Sigma^*} \leftrightarrow (<M>, <M_{\Sigma^*}>) \in L_{EQ}$.

COMPLEJIDAD COMPUTACIONAL

Ejercicio 6. Explicar informal y brevemente porqué el complemento de un lenguaje de P también está en P.

Ejercicio 7. Explicar por qué es incorrecta la siguiente prueba de que si $L \in NP$ entonces también $L^C \in NP$:
- Por hipótesis, existe una MTN M que acepta L en tiempo poly(n).
- Sea M´ una MTN que trabaja como M salvo que cuando M acepta M´ rechaza y viceversa (de esta manera tarda lo mismo que M).
- Como la MTN M´ acepta L^C en tiempo poly(n), entonces $L^C \in NP$.

Ejercicio 8. Un grafo tiene un camino de Hamilton de su vértice v_1 a su vértice v_m si tiene un camino de v_1 a v_m que pasa por el resto de los vértices sólo una vez.
- a) Expresar el problema mediante un lenguaje L_{CdeH}.
- b) Probar que $L_{CdeH} \in NP$.

Ejercicio 9. Se prueba que el lenguaje L_{CdeH} del ejercicio anterior es NP-completo.
- a) ¿Cuándo un lenguaje es NP-completo?
- b) Si existiese una reducción polinomial de L_{CdeH} a un lenguaje L de NP, ¿sería también L un lenguaje NP-completo? Justificar.

Ejercicio 10. Explicar por qué la clase temporal P está incluida en la clase espacial PSPACE.

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.

<u>Comentarios</u>

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

q1,1→q2,2 q2,2→q3,3 q3,3→q1,1
q1,B→q2

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 Σ^*. Explicar por qué cada uno de ellos es recursivo. <u>Ayuda</u>: 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? para ny

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. <u>Ayuda</u>: tener en cuenta una de las propiedades que estudiamos de las reducciones de lenguajes.

si L∈RE y L ∝ Lu y Lu ∈ R! L ∝ L2 ER

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). <u>Ayuda</u>: recordar que las MT se pueden ordenar.

generar códigos hasta encontrar el i, asignar el i / just validez

COMPLEJIDAD COMPUTACIONAL

Ejercicio 6. Explicar cuándo un lenguaje pertenece a:
(a) La clase P. acepta poly(n)
(b) La clase NP. verifica poly(n)
(c) La clase NPC.
NP y ∀L∈NP ∝ L

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. <u>Ayuda</u>: la solución en un sentido es similar a la que vimos para probar que CLIQUE está en NP. ver cert

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

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. si cond 1 ej 7 cond 2 satisfac

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

ya que por $c^{S(n)}$ termino en PTIME

porque en n pasos se recorren a lo sumo n celdas

CamScanner

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 (¬, ∧).** $p \wedge \neg p$

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

$\neg(p \to p)$

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

p → q ∨ r

"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"

¬q → ¬p

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

1. (t → u) h
2. (t → (u → t)) → ((t → u)→(t → t)) L2
3. (t → (u → t)) L1
4. ((t → u)→(t → t)) MP
5. (t → t) A

{t → u} (t → t)

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. si, es tanto

Ejercicio 4. Sea B una fbf cualquiera. ¿Es posible que se cumpla $\vdash_L B$ y al mismo tiempo B sea una contradicción? Justificar. no, tiene que ser tanto

Ejercicio 5. Dado el predicado M(x,y) que representa *"x es mayor que y"* y la función *f(x)* que representa *"x-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. $\exists x \forall y\, M(x, y)$

b) Todo número es mayor que su antecesor. $\forall x\, M(x, f(x))$

c) Para todo par de números existe uno que está entre ambos. $\forall x \forall y \exists z (M(x, z) \wedge M(z, y))$

* *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.

(∃x P(x) ∧ ∃x Q(x)) → (∃x (P(x) ∧ Q(x)))

P = par Q = impar

$x_1 = 2$ $x_2 = 3$

FUNDAMENTOS DE TEORÍA DE LA COMPUTACIÓN

Computabilidad y Complejidad Computacional. Examen 1 Año 2024. Licenciatura en Sistemas.

Plantel Docente: C. Pons, R. Rosenfeld, L. Mendoza, P. Dal Bianco, M. Manzín.

Comentario: El examen dura 2 horas y es a libro cerrado. Ser claros y breves. Aprovechar bien el tiempo, no dedicar mucho tiempo a ningún ejercicio en particular. Hay que hacer ejercicios de Computabilidad y de Complejidad Computacional. ¡Exitos!

COMPUTABILIDAD

Ejercicio 1.

a) Definir cuándo un lenguaje es recursivo.

b) Dar un ejemplo de un lenguaje recursivo y un ejemplo de un lenguaje no recursivo.

c) Si un lenguaje es recursivo, ¿también lo es su complemento? Justificar.

Ejercicio 2. Dadas dos MT M_1 y M_2 que reconocen lenguajes L_1 y L_2 de la clase RE, respectivamente, explicar cómo se puede construir una MT M que reconozca la intersección de L_1 con L_2. *Comentario: sólo dar la idea general.*

Ejercicio 3. Dado el lenguaje $D = \{w_i \mid M_i \text{ acepta } w_i\}$, considerando el orden canónico de las cadenas, explicar cómo se puede construir una MT M que lo reconozca. *Comentario: sólo dar la idea general.*

Ejercicio 4. Se cumple que todos los lenguajes de RE se reducen al lenguaje HP. Explicar por qué, si HP fuera recursivo, se cumpliría RE = R.

Ejercicio 5. Explicar por qué la función f: $\Sigma^* \rightarrow \Sigma^*$ no es una reducción del lenguaje L_U al lenguaje L_U^C: a todo par (<M>, w) le asigna el par (<M^C>, w), tal que <M^C> es como <M> salvo que sus estados finales están invertidos, es decir, donde aparece q_A en <M> aparece q_R en <M^C> y viceversa. *Comentario: para simplificar, asumir que los pares (<M>, w) de entrada siempre son sintácticamente correctos.*

COMPLEJIDAD COMPUTACIONAL

Ejercicio 6.

a) Definir cuándo un lenguaje pertenece a la clase P.

b) Dar un ejemplo de un lenguaje de P y un ejemplo de un lenguaje que no estaría en P, justificando por qué no estaría.

Ejercicio 7.

a) Definir cuándo un lenguaje pertenece a la clase NP.

b) Dar un ejemplo de un lenguaje de NP e indicar cómo serían los certificados de sus cadenas.

Ejercicio 8. Sea L_1 un lenguaje NP-completo y L_2 un lenguaje de NP. Explicar por qué, si existe una reducción polinomial de L_1 a L_2, se cumple que L_2 es NP-completo.

Ejercicio 9. Sea TAUT el lenguaje de las tautologías, es decir: TAUT = $\{\varphi \mid \varphi$ es una fórmula booleana sin cuantificadores satisfactible con toda asignación de valores de verdad$\}$.

a) Explicar por qué TAUT no estaría en la clase P.

b) Explicar por qué TAUT no estaría ni siquiera en la clase NP.

Ejercicio 10.

a) ¿Qué lenguajes se consideran tratables en la jerarquía espacial? Justificar por qué.

b) Explicar por qué una MT que ocupa espacio poly(n) no tiene por qué tardar tiempo poly(n).

c) Dada una MT M_1 que ocupa espacio S(n), explicar cómo se puede construir una MT M_2 que en espacio S(n) imprima una cadena con S(n) marcas X.