Parcial FTC - 2018

Ejercicio 1

La tesis de Church-Turing formula que todo dispositivo computacional físicamente realizable puede ser simulado por una máquina de Turing.

Ejercicio 2

$PARTICION = \{K = \{x_1, \ldots, x_n\} \mid los\ x_i\ son\ números\ naturales\ y\ existe\ una\ partición\ de\ K\ de\ la\ forma\ K_1 = \{y_1, \ldots, y_a\}\ \ y\ K_2 = \{z_1, \ldots, z_b\}\ que\ cumple\ que\ la\ suma\ de\ los\ números\ de\ K_1\ es\ igual\ a\ la\ suma\ de\ los\ números\ de\ K_2\}$

Ejercicio 3

Que ambos modelos de MT sean equivalentes significa que dada una MT de un modelo existe una MT equivalente del otro, es decir, existe una MT que acepta el mismo lenguaje pero es de un modelo distinto.

Ejercicio 4

Idea General: Podemos construir una MT M_{10} que acepte el lenguaje de las cadenas de 1 y 0 que empiezan con la subcadena 10 teniendo en cuenta que con que nos aseguremos que la cadena empieza de la forma deseada, luego podemos mantenernos en un estado genérico de lectura hasta llegar al Blanco y aceptar la cadena. El chequeo de la subcadena inicial lo podemos hacer usando 2 estados, uno para chequear el primer 1 y el otro para chequear el 0.

$M_{10} = (Q, \Gamma, \delta, q_0, q_A, q_R)$

$Q = (q_0, q_1, q_{lec}, q_A, q_R)$

- q_0 → chequea el primer 1.
- q_1 → chequea el 0 de la subcadena.
- q_lec → estado genérico de lectura.

$\Gamma = (1, 0, B)$

δ

- $\delta(q_0, 1) = \{(q_1, 1, R)\}$ → leímos el primer 1
- $\delta(q_1, 0) = \{(q_{lec}, 0, R)\}$ → leímos el 0 y completamos la subcadena

- $\delta(q_{lec}, 1) = \{(q_{lec}, 1, R)\}$ → lectura genérica
- $\delta(q_{lec}, 0) = \{(q_{lec}, 0, R)\}$ → lectura genérica
- $\delta(q_{lec}, B) = \{(q_A, B, S)\}$ → llegamos al final de la cadena correctamente

Ejercicio 5

Punto A

$R \subseteq RE$ se ve por definición de los 2 conjuntos, en R están todos los lenguajes que tienen una MT que los decide y para siempre para toda instancia del lenguaje ya sea positiva o negativa, en RE están todos los lenguajes que tienen una MT que los acepta, es decir, para siempre para toda instancia positiva del problema, en las instancias negativas puede parar y rechazar o loopear, todo lenguaje de R cumple la definición para pertenecer a RE ya que las MT que los deciden paran siempre para las instancias positivas del problema.

Punto B

$RE \subseteq \mathcal{L}$ se ve porque $\mathcal{L}$ es el conjunto de todos los lenguajes, ya sean lenguajes recursivos, recursivamente enumerables o no recursivamente enumerables, los lenguajes que pertenecen al conjunto RE son los recursivamente enumerables, por lo tanto, todo lenguaje perteneciente a RE pertenece a $\mathcal{L}$.

Punto C

$R \subseteq CO - RE$ se justifica de la siguiente forma, por definición, $L \in R$ sii $L^C \in R$, como vimos antes, $R \subseteq RE$ así que podemos decir que $L^C \in RE$ y si $L^C \in RE$ se cumple por definición que $(L^C)^C \in CO - RE$ siendo $(L^C)^C = L$, por lo tanto, afirmamos que $R \subseteq CO - RE$.

Punto D

Lenguaje $RE - R$

- $L_U = \{(< M >, w) \mid < M > \ acepta \ w\}$
  Lenguaje $CO - RE - R$
- $L_U^C = \{(< M >, w) \mid < M > \ rechaza \ w\}$

Ejercicio 6

M acepta la intersección entre los dos lenguajes debido a que, lógicamente, la intersección se ve como un AND lógico donde si la cadena w pertenece a los 2 lenguajes solo en ese caso la intersección se vuelve verdadera, analizando el comportamiento de M vemos que se resuelve

correctamente la intersección, **tomemos el rechazo o loop como un $False$ y la aceptación como un $True$:**

- Si M_1 rechaza o loopea ($False$) no nos hace falta analizar la otra parte de la intersección, M rechaza. *Esto es correcto.*
- Si M_1 acepta ($True$), ejecuta M_2:
  - Si M_2 rechaza o loopea ($False$) la intersección no se cumple, M rechaza. *Esto es correcto.*
    Si M_2 acepta ($True$) la intersección es verdadera ($True\ AND\ True$), M acepta. *Esto es correcto.*

Ejercicio 7

Para probar que $D \in RE$ nos basta con construir una MT que acepte el lenguaje y pare siempre para toda instancia positiva del problema, podemos hacerlo de la siguiente manera:

- Dada la entrada w_i de entrada, generará cadenas en el orden canónico hasta encontrar la cadena w idéntica a la w_i, de ahí sacará el índice que representa a la cadena de entrada en el orden canónico.
- Una vez obtenido el índice de la cadena de entrada generará códigos válidos de máquinas de Turing según el orden canónico hasta dar con la $i-ésima$ máquina de Turing según el orden canónico.
- Una vez encontrada la $i-ésima$ máquina de Turing según el orden canónico, M_D simulará la cadena w sobre la máquina M_i:
  - Si M_i para y acepta, M_D acepta.
  - Si M_i para y rechaza, M_D acepta.
  - Si M_i loopea, M_D loopea.

Ejercicio 8

Punto A

Que f sea una reducción de L_1 a L_2 significa que f es una función que cumple 2 propiedades:

- Es *total computable*, es decir, está definida para todo el alfabeto y siempre genera un resultado, no loopea (es computable por una M_f).
- Es *correcta*, se cumple siempre que si $w \in L_1 \rightarrow f(w) \in L_2$ y si $w \notin L_1 \rightarrow f(w) \notin L_2$.

Punto B

Si $L_2 \in R$ por propiedad de las reducciones "$L_1 \leq L_2$ entonces $L_2 \in R \rightarrow L_1 \in R$" podemos afirmar que $L_1 \in R$.

Punto C

Si $L_1 \notin RE$ por propiedad de las reducciones "$L_1 \leq L_2$ entonces $L_1 \notin RE \rightarrow L_2 \notin RE$" podemos afirmar que $L_2 \notin RE$.

Ejercicio 9

Punto A

- MT general: Sobre la cinta de input el cabezal se puede mover hacia la derecha, izquierda o quedarse en el lugar.
- Autómata Finito: Sobre la cinta de input el cabezal solo puede moverse hacia la derecha.
- Autómata con Pila: Sobre la cinta de input el cabezal solo puede moverse hacia la derecha.

Punto B

- MT general: Puede escribir sin restricciones sobre la cinta de input y cualquier otra cinta.
- Autómata Finito: Solo cuenta con una cinta de input de sólo lectura, no hace uso de citas adicionales.
- Autómata con Pila: Cuenta con una cinta de input de sólo lectura y una cinta adicional que puede escribir libremente (la escribe representando una pila).

Punto C

- MT general: Para según sus estados finales.
- Autómata Finito: Para al llegar al Blanco al final de la cadena recibida en la cinta de input.
- Autómata con Pila: Para al llegar al Blanco al final de la cadena recibida en la cinta de input.

Punto D

- MT general: Acepta o Rechaza según sus estados finales.
- Autómata Finito: Acepta o Rechaza según sus estados finales una vez haya parado, si paró y está en un estado final, acepta, sino, rechaza.
- Autómata con Pila: Acepta o Rechaza según el estado de su cinta adicional "pila" cuando para, si la misma está vacía al parar, acepta, sino, rechaza.

Ejercicio 10

Que un lenguaje L pertenezca a la clase $TIME(T(n))$ significa que **L es decidido por una MT M en un tiempo de a lo sumo orden $O(T(n))$.**

Ejercicio 11

La **Tesis Fuerte de Church-Turing formula** "Si un lenguaje L es decidible en tiempo polinomial por un dispositivo computacional físicamente realizable, también es decidible en tiempo polinomial por una máquina de Turing".

Ejercicio 12

Para probar que el Problema de la Partición pertenece a NP vamos a demostrar que para todo K que pertenece al Problema de la Partición existe un **certificado sucinto** (de tamaño polinomial con respecto a la entrada) **verificable en tiempo polinomial por una máquina de Turing verificadora del problema** de la siguiente forma:

- Un *certificado sucinto* puede ser 2 particiones K_1 y K_2 del conjunto K cuyos elementos sean números naturales que cumplan la condición de que el total de todos los elementos de K_1 sumados es igual al total de todos los elementos de K_2 sumados. *El certificado es sucinto ya que en el peor de los casos, el tamaño de los 2 subconjuntos sería K, lo cual es polinomial con respecto a la entrada.*
- *La verificación del certificado también se da en tiempo polinomial:*
  - Debemos verificar que todo elemento de cada subconjunto K_1 y K_2 existe en el conjunto K y que $|K_1| + |K_2| = |K|$, esto en el peor de los casos representa un orden $O(|K| \cdot |K|) = O(|K|^2)$.
  - Por enunciado, la suma de dos números se realiza en tiempo polinomial por lo que el sumar todos los elementos de cada subconjunto y luego comparar los totales también se va a dar en tiempo polinomial, esto representa un orden de $O(|K|)$.
    La verificación se termina dando con un orden $O(|K|^2) + O(|K|) = O(|K|^2)$, lo que es polinomial.

Ejercicio 13

Para probar que $L_2 \in NP - completo$ tenemos que probar 2 cosas:

- L_2 pertenece a NP.
- L_2 es $NP - difícil$, es decir, para todo $L_i \in NP$ existe una reducción polinomial $L_i \leq_p L_2$.

Dado que existe una reducción polinomial $L_2 \leq_p L_1$ siendo $L_1 \in NP - completo$ por propiedad de las reducciones como vimos anteriormente podemos afirmar que $L_2 \in NP$ ya que $L_1 \in NP$. **De esta forma probamos que $L_2 \in NP$.**

Dado que $L_1 \in NP - completo$, sabemos que L_1 es $NP - difícil$, por lo tanto, para todo $L_i \in NP$ existe una reducción polinomial $L_i \leq_p L_1$, pero como sabemos que existe una reducción polinomial $L_1 \leq L_2$ por propiedad de la transitividad en las reducciones polinomiales "$L_i \leq_p L_1$ y $L_1 \leq_p L_2$ entonces $L_i \leq_p L_2$" **podemos afirmar que L_2 es $NP - difícil$.**

De esta forma demostramos que $L_2 \in NP - completo$.

Ejercicio 14

- **Paso III:** Esto se cumple por *definición de la clase $CO - NP$*, $L \in CO - NP$ sii $L^C \in NP$, al saber que $L_2 \in CO - NP$ se cumple que $L_2^C \in NP$.
- **Paso IV:** Esto se cumple por *definición de la clase $NP - completo$*, al $L_1 \in NP - completo$, L_1 es $NP - difícil$, es decir, para todo $L_i \in NP$ existe una reducción polinomial $L_i \leq_p L_1$. Como $L_2^C \in NP$ existe la reducción polinomial $L_2^C \leq_p L_1$.
- **Paso V:** Se cumple por *propiedad de las reducciones polinomiales* "$L_1 \leq_p L_2$ sii $L_1^C \leq_p L_2^C$". Como sabemos que existe la reducción polinomial $L_2^C \leq_p L_1$ debe existir la reducción polinomial $L_2 \leq_p L_1^C$.
- **Paso VI:** Se cumple por *propiedad de las reducciones polinomiales* "$L_1 \leq L_2$ entonces $L_2 \in NP \rightarrow L_1 \in NP$", como $L_1 \in CO - NP$ (*información del paso I*), $L_1^C \in NP$, esto hace que $L_2 \in NP$.

Ejercicio 15

CONSULTAR CON RICARDO.