# Practica 7

1) Construya una MTN que genere de manera no determinística todos los números de 8 bits. Es decir que, dado cualquier número, alguna computación de la máquina lo generará. ¿Cuántos movimientos hace la máquina?

   M=<Q, Σ, Γ, Δ, q0, qA, qR>, Δ: Q x Γ → r ((Q U {qA,qR}) x Γ x {D, I, S})

   Δ (q0, B) = {(q1, 0, D), (q1, 1, D)}
   Δ (q1, B) = {(q2, 0, D), (q2, 1, D)}
   Δ (q2, B) = {(q3, 0, D), (q3, 1, D)}
   Δ (q3, B) = {(q4, 0, D), (q4, 1, D)}
   Δ (q4, B) = {(q5, 0, D), (q5, 1, D)}
   Δ (q5, B) = {(q6, 0, D), (q6, 1, D)}
   Δ (q6, B) = {(q7, 0, D), (q7, 1, D)}
   Δ (q7, B) = {(qA, B, S), (qA, B, S)}

   Esta maquina realiza siempre 8 movimientos. Todas las otras combinaciones de estado símbolo que faltan llevan al estado qR

2) Sean L1 y L2, dos lenguajes definidos sobre {0,1}*
   $L1 = \{0^n 1 | n \geq 0\}$
   $L2 = \{1^n 0 | n \geq 0\}$

   a) Construya una MTN M tal que L(M)= L1 ∪ L2

   M=<Q, Σ, Γ, Δ, q0, qA, qR>, Δ: Q x Γ → r ((Q U {qA,qR}) x Γ x {D, I, S})

   Δ (q0, x) = {(q1, x, S), (q2, x, S)} para todo x ∈ Γ

   Δ (q1, 0) = {q1, 0, D)}
   Δ (q1, 1) = {q3, 1, D)}

   Δ (q2, 1) = {q2, 1, D)}
   Δ (q2, 0) = {q3, 0, D)}

   Δ (q3, B) = {qA, B, S)}

   Todas las otras combinaciones de estado símbolo que faltan llevan al estado qR

   b) Describa la traza de ejecución para las entradas w1=001 y w2= 1101

      w1 = 001

      $q_0 001 \vdash q_1 001 \vdash 0 q_1 01 \vdash 00\ q_1 1 \vdash 001 q_3 B \vdash 001 q_A B$
      $\vdash q_2 001 \vdash 0\ q_3 01 \vdash 0\ q_R 01$

      w2= 1101

$q_0 1101 \vdash q_1 1101 \vdash 1q_3 101 \vdash 1q_R 101$
$\vdash q_2 1101 \vdash 1q_2 101 \vdash 11q_2 01 \vdash 110q_3 1 \vdash 110q_R 1$

3) ¿La reducción polinomial posee las siguientes propiedades? Justifique
   a) Reflexiva

      ¿ L1 $\alpha$p L1?

      Si, se puede construir una MTD Mf que computa la función de reducibilidad de identidad en un tiempo polinomial.

   b) Simétrica

      ¿L1 $\alpha$p L2 $\Rightarrow$ L2 $\alpha$p L1?

      No, o al menos no por ahora. Contraejemplo: L1 $\in$ P y L2 $\in$ NP, se puede construir una MTD que compute la función de reducibilidad en n pasos (tiempo polinomial) de L1 a L2, pero no se puede construir una MTD que realice lo mismo pero de L2 a L1, porque esto significaría que P = NP (ya que dado que L1 $\in$ P, por teorema L2 tambien $\in$ P) y eso todavía aun no se ha demostrado.

   c) Antisimétrica

      ¿L1 $\alpha$p L2 $\wedge$ L2 $\alpha$p L1 $\Rightarrow$ L1 = L2?

      No. Contraejemplo: L1 = { 0^n 1 / n >= 0 } y L2 = { 1^n 0 / n >= 0 } se puede hacer L1 $\alpha$p L2 y L2 $\alpha$p L1 pero sabemos por definición que L1 $\neq$ L2.

   d) Transitiva

      ¿L1 $\alpha$p L2 $\wedge$ L2 $\alpha$p L3 $\Rightarrow$ L1 $\alpha$p L3?

      Si, se puede realizar una reducción de L1 a L2 con una MTD en n pasos (tiempo polinomial) y una reducción de L2 a L3 con una MTD en k pasos (tiempo polinomial), por lo que para reducir con una MTD L1 a L3 se van a requerir n * k pasos (tiempo polinomial). Esto ultimo es sencillo, realizando una MTD que junte ambas MTD que realizan las reducciones de L1 $\alpha$p L2 y L2 $\alpha$p L3.

4) ¿Es cierto que si dos lenguajes L1 y L2 son NPC entonces L1 $\alpha$p L2, y también L2 $\alpha$p L1? Justifique su respuesta.

   Sabemos que como L1 $\in$ NPC (NP-Completo) entonces, por definición, L1 $\in$ NPH y L1 $\in$ NP.

   Que L1 $\in$ NPH significa que para todo L $\in$ NP se cumple L $\alpha$p L1.

Como se sabe que L2 $\in$ NPC, se sabe que L2 $\in$ NP (por definición de NPC), por lo tanto, por lo anterior dicho se puede afirmar que L2 $\alpha$p L1 (definición de NPH). Se puede demostrar análogamente para L1 $\alpha$p L2