MA1101-2 Introducción al Álgebra
Profesor: Leonardo Sánchez C.
Auxiliar: Marcelo Navarro

Desafios Control 3

P1. **[Relaciones y funciones]** Sean X, Y conjuntos y $f : X \to Y$ una función.

a) Sea $\mathcal{R}$ una relación de equivalencia definida sobre Y. Se define en X la relación preimagen de $\mathcal{R}$ que notamos $f^{-1}(\mathcal{R})$ por:

$$x_1 \; f^{-1}(\mathcal{R}) \; x_2 \iff f(x_1) \; \mathcal{R} \; f(x_2)$$

1) Probar que $f^{-1}(\mathcal{R})$ es una relación de equivalencia en X
2) Probar que $[x]_{f^{-1}(\mathcal{R})} = f^{-1}([f(x)]_{\mathcal{R}})$ para cada $x \in X$
Indicación: Notar que hay que probar una igualdad de conjuntos y que $[f(x)]_{\mathcal{R}}$ es un subconjunto de Y.

b) Suponga ahora que $\mathcal{R}'$ es una relación de equivalencia en X y que f es epiyectiva. Definimos en Y la relación imagen por:

$$y_1 \; f(\mathcal{R}') \; y_2 \iff \exists x_1, x_2 \in X, \; f(x_1) = y_1 \quad \wedge \quad f(x_2) = y_2 \quad \wedge \quad x_1 \; \mathcal{R}' \; x_2$$

1) Probar que si f es inyectiva entonces $f(\mathcal{R}')$ es de equivalencia y además pruebe que $\mathcal{R}' = f^{-1}(f(\mathcal{R}'))$, es decir, $\mathcal{R}'$ es igual a la relación preimagen de $f(\mathcal{R}')$
2) ¿Que ocurre si f no fuera inyectiva?
Propuesto: Defina f tal que no sea inyectiva y $f(\mathcal{R}')$ no esa de equivalencia

P2. **[Aplicación de Relaciones: Teoria de la Computación]**

El objetivo de esta pregunta es ver una de las aplicaciones de las clases de equivalencia en el campo de la teoría de la computación, particularmente en autómatas finitos y maquinas de Turing, que nos permiten decidir si los problemas se pueden resolver o no. Por ejemplo, gracias a estas herramientas podemos demostrar que no existe algoritmo que determine de manera correcta si un programa arbitrario se detendrá, una vez sea ejecutado. Este problema es conocido como Halting Problem.

Definición 1 (Símbolo). *Se define un símbolo como un carácter cualquiera. Por ejemplo:* $a, 1, \heartsuit$, *etc.*

Definición 2 (String). *Se define un string o palabra como una secuencia de símbolos o bien, un símbolo. Por ejemplo:* $a, b, hola, aabba, 00010, 101, \heartsuit\heartsuit 0a$, *etc. También se define el string vació 'ε', como aquel string que simplemente no hace nada a dentro de una secuencia. Ejemplo:* $'hola\varepsilon' = 'hola'$.

Definición 3 (Concatenación). *Dados dos strings* $'a'$ *y* $'b'$, *se define la concatenación entre* $'a'$ *y* $'b'$ *como* $'a \cdot b'$ *o bien* $'ab'$, *donde simplemente se ha puesto al final de* $'a'$, *el string* $'b'$. *Por ejemplo: la concatenación entre* $'hol'$ *y* $'a'$ *es* $'hola'$.

Definición 4 (Alfabeto). *Se define un alfabeto como un conjunto de símbolos, lo denotaremos por* Σ.

Definición 5. *Se define* Σ^* *como el conjunto de todas las palabras o strings que se pueda formar a partir de los símbolos en* Σ.
Por ejemplo si $\Sigma = \{a, b\}$ *entonces*

$$\Sigma^* = \{\varepsilon, a, b, aa, bb, ab, ba, aaa, \ldots, ababbabb, \ldots\}$$

Este conjunto se define recursivamente por.

- $\varepsilon \in \Sigma^*$
- $w \in \Sigma^* \; y \; a \in \Sigma \Rightarrow wa \in \Sigma^*$

Definición 6 (Lenguaje). *Definimos un Lenguaje* L, *como un subconjunto de* Σ^*, *es decir,* $\emptyset \subseteq L \subseteq \Sigma^*$.

(Ahora empieza la pregunta) Sea el alfabeto $\Sigma = \{0, 1\}$ y sea $L \subseteq \Sigma^*$ un lenguaje cualquiera. se define la relación $\equiv_L$ sobre Σ^* tal que

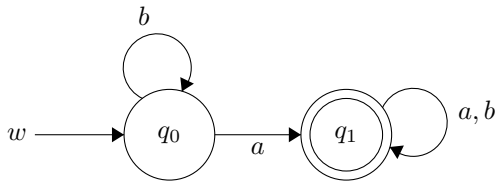
$$x \equiv_L y \iff [\forall z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L]$$

Es decir, en $\equiv_L$ viven todas las parejas de palabras que les falta lo mismo para vivir en L.

a) Demuestre que $\equiv_L$ es una relación de equivalencia.

b) Considere ahora $L = \{w \in \Sigma^* : w \text{ contiene 001 como un substring }\}$. Calcule $\Sigma^* / \equiv_L$.

c) Dibuje una maquina de estados y transiciones (que llamaremos autómatas) que acepte al lenguaje L, donde una palabra w se ira leyendo de izquierda a derecha, caracter por caracter, y dependiendo del caracter leido se escoge un estado siguiente de la maquina (esto lo definimos como una transición). La maquina termina de procesar cuando se termina el string, y se dice que la maquina acepta un string si la maquina termina en un estado de aceptación. Por ejemplo, la maquina para el lenguaje que contiene al menos un a es:



Donde el estado de aceptación es q_1. Notar que cualquier palabra que contiene una a se quedará por siempre en q_1 y una palabra que no contiene a siempre se quedara en q_0 por lo que no será aceptada.

Hint: piense en las clases de equivalencia que encontro en la parte *b*)