Métodos Lógicos para Ciencia de la Computación - CC51N
Tarea 3

1. Usando juegos de Ehrenfeucht-Fraïssé, demuestre que no existe una oración ϕ de la lógica de primer orden sobre el vocabulario de los grafos tal que para todo grafo finito G, $G \models \phi$ si y solo si G es conexo.

2. Sea σ un vocabulario que solo contiene relaciones. Dadas dos σ-estructuras $\mathcal{A}$ y $\mathcal{B}$, definimos su producto cartesiano $\mathcal{A} \times \mathcal{B}$ como la siguiente σ-estructura: El dominio de $\mathcal{A} \times \mathcal{B}$ es $A \times B$, donde A es el dominio de $\mathcal{A}$ y B es el dominio de $\mathcal{B}$. Además, para cada símbolo m-ario R en σ, la interpretación del símbolo R en $\mathcal{A} \times \mathcal{B}$ contiene todas las tuplas $((a_1, b_1), \ldots, (a_m, b_m)) \in (A \times B)^m$ tal que $(a_1, \ldots, a_m)$ pertenece a la interpretación de R en $\mathcal{A}$ y $(b_1, \ldots, b_m)$ pertenece a la interpretación de R en $\mathcal{B}$.

   Sean $\mathcal{A}_1, \mathcal{A}_2, \mathcal{B}_1, \mathcal{B}_2$ estructuras sobre el vocabulario σ. Demuestre que para todo $k \geq 0$, si $\mathcal{A}_1 \equiv_k \mathcal{B}_1$ y $\mathcal{A}_2 \equiv_k \mathcal{B}_2$ entonces debe ser el caso que $\mathcal{A}_1 \times \mathcal{A}_2 \equiv_k \mathcal{B}_1 \times \mathcal{B}_2$.

3. Sea Σ un alfabeto finito. Los strings sobre alfabeto Σ pueden representarse como estructuras finitas sobre el alfabeto σ_Σ que contiene un símbolo binario $<$ y símbolos unarios P_a para cada $a \in \Sigma$. Un string $s \in \Sigma^*$ de largo n puede entonces representarse como la estructura $\mathcal{M}_s$ sobre alfabeto σ_Σ tal que el dominio de $\mathcal{M}_s$ es $\{1, \ldots, n\}$, el símbolo $<$ se interpreta como el orden natural en $\{1, \ldots, n\}$, y para cada $a \in \Sigma$, P_a contiene el conjunto de aquellas posiciones del string donde la letra a ocurre.

   Definimos a las expresiones *star-free* sobre Σ inductivamente de la siguiente forma:

   - $\emptyset$, ϵ, y a ($a \in \Sigma$) son expresiones star-free.
   - Si R es expresión star-free, entonces también lo es $\bar{R}$.
   - Si R y S son expresiones star-free, entonces también lo son $R \cup S$ y RS.

   Interpretamos cada expresión star-free R como un lenguaje $L(R) \subseteq \Sigma^*$ de la siguiente forma:

   - $L(\emptyset) = \emptyset$, $L(\epsilon)$ es el lenguaje que solo contiene al string vacío, y $L(a) = \{a\}$, para cada $a \in \Sigma$.
   - $L(\bar{R}) = \Sigma^*/L(R)$.
   - $L(R \cup S) = L(R) \cup L(S)$ y $L(RS) = L(R) \cdot L(S)$.

   Demuestre lo siguiente:

   a) Para cada expresión star-free R sobre Σ es posible construir expresión regular R' sobre Σ tal que $L(R) = L(R')$, donde $L(R')$ es el lenguaje regular asociado con la expresión regular R'.

   b) Para cada expresión star-free R sobre Σ es posible construir una oración ϕ de la lógica de primer orden sobre el vocabulario σ_Σ tal que, para cada string $s \in \Sigma^*$, $s \in L(R) \Leftrightarrow \mathcal{M}_s \models \phi$.

   c) Para cada oración ϕ de la lógica de primer orden sobre vocabulario σ_Σ es posible construir una expresión star-free R sobre Σ tal que, para cada string $s \in \Sigma^*$, $s \in L(R) \Leftrightarrow \mathcal{M}_s \models \phi$.