

UNIVERSIDAD AUTONOMA METROPOLITANA Cuajimalpa
Inteligencia artificial

Tarea 3

Nombre:

Matrícula:

Fecha de entrega: 31 de octubre de 2025.

El siguiente grupo de problemas involucra funciones que operan con conjuntos. Un conjunto es un colección de elementos, cada uno de los cuales se dice que es un miembro del conjunto. Un conjunto puede ser representado como una lista, con cada elemento del conjunto representado por un átomo. Cada átomo aparece solo una vez en la lista, y no se le asigna ningún significado al orden de aparición de los átomos en la lista. Supongase que representamos conjuntos como listas lo cual implica que ningún elemento se repite. Implemente las rutinas correspondientes en cada problema que aparece a continuación.

1. Sabemos de la teoría de conjuntos que la operación de **Unión** entre dos conjuntos A y B se define como $A \cup B = \{x | x \in A \vee x \in B\}$. Implemente una función en **COMMONLISP** utilizando **defun**, **cons**, **member** y recursividad, que realice la operación $A \cup B$.
2. Sabemos de la teoría de conjuntos que la operación de **Intersección** entre dos conjuntos A y B se define como $A \cap B = \{x | x \in A \wedge x \in B\}$. Implemente una función en **COMMONLISP** utilizando **defun**, **cons**, **member** y recursividad, que realice la operación $A \cap B$.
3. Sabemos de la teoría de conjuntos que la operación de **Diferencia** entre dos conjuntos A y B se define como $A - B = \{x | x \in A \wedge x \notin B\}$. Implemente una función en **COMMONLISP** utilizando **defun**, **cons**, **member** y recursividad, que realice la operación $A - B$.
4. En **COMMONLISP** existen instrucciones llamados predicados que sirven para comprobar si un objeto o variable es un átomo, una lista, un número como se ilustra a continuación.

```
>(setf lista '( a b c ))
```

```
(A B C)
```

```
>(listp lista)
```

```
T
```

```
>(numberp 5)
```

```
T
```

```
>(listp 5)
```

```
NIL
```

```
>(numberp lista)
```

```
NIL
```

Haciendo uso de la rutina que implemente para la realizar la operación de intersección entre dos conjuntos defina una función **INTERSECCIONP**, que sea un predicado que prueba si dos conjuntos tienen cualquier elemento en común. Devolverá **NIL** si los dos conjuntos son disjuntos.

5. Sabemos de la **Teoría de Conjuntos** que un conjunto A es subconjunto de un conjunto B , lo cual se escribe matemáticamente como $A \subseteq B$, y que dicha relación entre conjuntos se define formalmente como sigue:

$$A \subseteq B \Leftrightarrow \forall x \in A \rightarrow x \in B \quad (1)$$

Defina el predicado **SUBCONJUNTOP** en **COMMONLISP** que regrese **T** si todos los elementos del conjunto representado por la lista **lista1** están presentes en el conjunto representado por la lista **lista2**

6. Sabemos de la **Teoría de Conjuntos** que un conjunto A es igual a un conjunto B si se cumple la siguiente relación entre ellos

$$(A \subseteq B) \wedge (B \subseteq A) \quad (2)$$

Defina una función en **COMMONLISP** llamada **MISMOCONJP**, que sea un predicado que prueba si dos conjuntos contienen los mismos elementos. Tenga en cuenta que las dos listas que representan los conjuntos pueden estar sus elementos ordenados de forma diferente.

7. Defina una función en **COMMONLISP** que dé el n -ésimo elemento de una lista, es decir que tenga como parámetro un número **x** y una lista de elementos **lista** y que dé el elemento de la **lista** que se encuentre en la posición **x**.
8. Defina una función en **COMMONLISP** que dado un elemento y una lista regrese la posición en la lista donde encontró la primera ocurrencia de dicho elemento.
9. Defina una función en **COMMONLISP** que inserte el elemento **e** en la posición **p** de la lista **lista**.

10. Defina una función en **COMMONLISP** que inserte el elemento **e** en la posición **p** de la lista **lista**.
11. Defina una función en **COMMONLISP** que sustituya un elemento de una lista **lista** en la posición **p** por el nuevo elemento **x**.
12. Defina una función en **COMMONLISP** que borre un elemento de una lista **lista** en la posición **p** y devuelva la nueva lista resultante.