Programación para la Inteligencia Artificial
Tarea 4

Dr. Alejandro Guerra-Hernández

Universidad Veracruzana
Instituto de Investigaciones en Inteligencia Artificial
Campus Sur, Calle Paseo Lote II, Sección Segunda No. 112,
Nuevo Xalapa, Xalapa, Ver., México 91097

aguerra@uv.mx
https://www.uv.mx/personal/aguerra

18 de noviembre de 2024

1. Implemente en Lisp las siguientes operaciones sobre conjuntos representados como listas. [15 puntos]

   - Subconjunto:

```
> (subset '(1 3) '(1 2 3 4))
true.
> (subset '() (1 2)).
true.
```

   - Intersección:

```
> (inter '(1 2 3) '(2 3 4)).
(2 3)
```

   - Unión:

```
> (union '(1 2 3 4) '(2 3 4 5))
(1 2 3 4 5)
```

   - Diferencia:

```
> (dif '(1 2 3 4) '(2 3 4 5))
(1)
> (dif '(1 2 3) '(1 4 5))
(2 3)
```

2. Escriba un programa en que elimine todas las ocurrencias de un elemento en una lista. Por ejemplo:

```
> (eliminar 3 '(1 3 2 4 5 3 6 7))
(1 2 4 5 6 7)
```

   Explique brevemente cómo es que Lisp evalúa esta expresión. [10 puntos]

3. Implemente una función en Lisp que dada una lista de átomos, regresa las posibles permutaciones de sus miembros. [10 puntos]

```
> (perms '(1 2 3))
((1 2 3) (1 3 2) (2 3 1) (2 1 3) (3 1 2) (3 2 1))
```

4. ¿Qué diferencias importantes puede señalar entre la implementación de los ejercicios anteriores y la llevada a cabo con Prolog. Sean concisos en la respuesta. [10 puntos]

5. Lea el capítulo 22 (*LOOP for Black Belts*) del libro de Peter Seibel, Practical Common Lisp. Utilice la macro `loop` para resolver alguno de los ejercicios propuestos en esta tarea. [15 puntos]

6. Defina una macro `repeat` que tenga el siguiente comportamiento. Evidentemente, la expresión que se repite puede ser cualquier expresión válida en Lisp. [15 puntos]:

```
> (repeat 3 (print 'hi))
HI
HI
HI
NIL
```

7. La siguiente función me permite definir una entrada en un registro de mis libros:

```
(defun crea-libro (titulo autor ed precio)
   (list :titulo titulo :autor autor :ed ed :precio precio))
```

   Puedo usar una variable global como `*db*` para llevar un registro de entradas como sigue:

```
(defvar *db* nil)

(defun agregar-reg (libro) (push libro *db*))
```

De forma que:

```
> (agregar-reg (crea-libro "Pericia Artificial" "Alejadro Guerra" "UV" 90.50))
((:TITULO "Pericia Artificial" :AUTOR "Alejadro Guerra" :ED "UV" :PRECIO 90.5))
```

Agregue más entradas al registro y escriba una función con ayuda de `format` (Ver capítulo 18 del libro de Seibel) que despliegue las entradas como sigue [15 puntos]:

```
> (listado-db)
TITULO:   Pericia Artificial
AUTOR:    Alejadro Guerra
ED:       UV
PRECIO:   90.50
```

8. Defina una función para recuperar una entrada en el registro buscando por autor [10 puntos].