



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

## FACULTAD DE CIENCIAS

### LÓGICA COMPUTACIONAL



Profesora: ESTEFANÍA PRIETOS LARIOS  
Ayudante: HERNÁNDEZ OLVERA MAURICIO E.  
Ayudante Lab: SALAZAR GONZÁLEZ EDWIN MAX

Fecha de Entrega: Jueves 4 de Abril, 2019. 13:59

## PRÁCTICA 2. Parte 2.

### PROLOG

#### OBJETIVOS

- Reforzar los conceptos básicos del lenguaje lógico PROLOG.
- Se espera que el alumno adopte los conceptos sobre manejo de listas en *prolog*.

#### INSTRUCCIONES

Su misión, si deciden aceptarla es:

Descargar el archivo *Practica2p2.pl* y resolver al menos 10 de los ejercicios definidos sobre éste.

#### EJERCICIOS

1.- Crea la regla **getnth**

Recibe tres parámetros: E, N, L

La regla será satisfactoria cuando el elemento E esté en el índice N de la lista L.

```
| ?- getnth(E, 4, [S,a,v,e,T,h,e,P,a,r,t,i,e,s,!]).  
E = T ? ;  
no
```

## 2.- Crea la regla **getIndex**

Recibe tres parámetros: E, L, N

La regla será satisfactoria cuando el elemento E esté en el índice N de la lista L.

```
| ?- getIndex(E, [S,a,v,e,T,h,e,P,a,r,t,i,e,s,!], 4).  
E = T ? ;  
no
```

## 3.- Crea la regla **num\_ocurr**

Recibe tres parámetros: E,L,R

La regla será satisfactoria cuando el elemento E ocurra R veces en la lista L.

```
| ?- num_ocurr(a, [h,o,l,a,a,a,a,a,a,a], R).  
R = 8 ? ;  
no
```

## 4.- Crea la regla **concatena**

Recibe tres parámetros: L1, L2, LF.

La regla será satisfactoria cuando L1 se concatene con L2, obteniendo la lista LF.

```
| ?- concatena([1,2,3,4], [5,6,7,8], Lf).  
Lf = [1,2,3,4,5,6,7,8] ? ;  
yes
```

## 5.- Crea la regla **deleteL**

Recibe tres parámetros: I,L,R.

La regla será satisfactoria cuando se elimine al elemento en la I-ésima posición de la lista L. La lista resultante será guardada en la variable R.

```
| ?- deleteL(1, [u,n,a,m], Z).  
Z = [u,a,m] ? ;  
no
```

#### 6.- Crea la regla **sumaLista**

Recibe dos parámetros: L, R.

La regla será satisfactoria cuando R tenga el resultado de sumar los elementos de la lista L

```
| ?- sumaLista([1,2,3,4], W).  
W = 10  
yes
```

#### 7.- Crea la regla **acumulador**

Recibe dos parámetros: L, K.

La regla será satisfactoria cuando K tenga el resultado de sumar los elementos anteriores al i-ésimo índice.

K es una lista.

Se recomienda el uso de una regla auxiliar.

```
| ?- acumulador([1,2,3,4,5,6], T).  
T = [1,3,6,10,15,21]  
yes
```

#### 8.- Crea la regla **update**

Recibe cuatro parámetros: N, E, XS, R.

La regla actualiza el índice N de la lista XS con el elemento E.

El resultado se guarda en la variable R.

```
| ?- update(5, s, [G,i,m,n,a,c,i,o], W).  
W = [G,i,m,n,a,s,i,o] ? ;  
no
```

#### 9.- Crea la regla **subset**

Recibe dos parámetros: L, K.

La regla será satisfactoria cuando K sea subconjunto de L.

```
| ?- subset([1,2,3,4,5,6,7,8,9], [2,4,6,8]).  
True ? ;
```

#### 10.- Crea la regla **union**

Recibe tres parámetros: A, B, C

La regla será satisfactoria cuando C sea el resultado de unir A con B.

En otras palabras, se debe cumplir  $C = A+B$ .

```
| ?- union([1,2,3,4,5], [6,7,8,9],D).  
D = [1,2,3,4,5,6,7,8,9]  
yes
```

#### 11.- Crea la regla **splitL**

Recibe cuatro parámetros: L, N, L1, L2

La regla será satisfactoria cuando se cumpla que la lista L fue dividida en dos listas (L1 y L2), partiéndola en el índice N.

```
| ?- splitL ([0,1,2,3,4,5,6,7,8,9], 5, U, V).  
U = [0,1,2,3,4,5]  
V = [6,7,8,9] ? ;  
no  
  
| ?- splitL ([0,1,2,3,4,5,6,7,8,9], 0, U, V).  
U = [0]  
V = [1,2,3,4,5,6,7,8,9] ? ;  
no
```

#### 12.- Crea la regla **drop**

Recibe tres parámetros: N, L, R.

La regla será satisfactoria cuando se cumpla la propiedad de que se eliminó cada N-ésimo elemento de la lista L.

La lista resultante se guarda en la variable R.

Nota: Borra **cada** n índices, no borra solamente el n-ésimo elemento.

Nota2: Se sugiere usar una regla auxiliar.

```
| ?- drop (2, [1,2,3,4,5,6], T).  
T = [1,3,5] ? ;  
no
```

## REGLAS DEL JUEGO:

- ✓ Respetar las firmas de las reglas.
- ✓ **Todas** las firmas deberán estar documentadas.
- ✓ La legibilidad y documentación tendrá un impacto sobre la calificación de la práctica.
- ✓ La práctica se podrá realizar en equipos de a lo más 3 personas.
- × “Comprar películas piratas se ve mal, pero ustedes pirateándose la práctica de su amig@ se ven mucho peor”. Cualquier plagio de prácticas será evaluado con 0, sin hacer indagaciones.
- × Cualquier copia de internet, sin entender el código **será 0 sobre la práctica**. Así se haya copiado solo en un ejercicio.
- × Cualquier práctica entregada posterior a la fecha límite no será tomada en cuenta.

Se deberá contar con un directorio cuyo nombre sea Practica2. Dentro del directorio se debe tener:

- README.txt, donde se incluya número de cuenta y comentarios sobre la práctica.
- Practica2p2.pl, script requerido para ésta práctica.

Comprimir el directorio con el formato ApellidoNombreP2p2. Comprimir con extensión .tar.gz o .zip

Si la práctica se realiza en equipo, agregar el nombre y número de cuenta de los integrantes en el README.txt

Enviar la práctica al correo [ciclomax9@ciencias.unam.mx](mailto:ciclomax9@ciencias.unam.mx) con el asunto [LC-Apellido-Nombre-P2p2].

*Suerte*

*"¿Los índices de los arrays deberían comenzar en 0 o en 1?  
Mi propuesta neutral de usar 0.5 fue rechazada, en mi opinión,  
sin la debida consideración"  
-- Stan Kelly-Bootle*