

Práctico 9: Estructuras Incompletas. Gramáticas.

Ejercicio 1

Escriba los siguientes predicados sobre listas de diferencias.

Para las listas de diferencias utilice la representación mediante términos estructurados de la forma: $X-Xr$

l_Id (L, LD)	LD es una lista de diferencias equivalente a la lista L .
ld_l (LD, L)	L es la lista equivalente a la lista de diferencias LD .
append_Id (A, B, C)	C es la lista de diferencia equivalente a concatenar las listas de diferencias A y B .
inserta_Id ($L1, X, L2$)	$L2$ es la lista de diferencias resultante de insertar el elemento X al comienzo de la lista de diferencias $L1$.
insertz_Id ($L1, X, L2$)	$L2$ es la lista de diferencias resultante de insertar el elemento X al final de la lista de diferencias $L1$.
rotacion_Id ($L1, L2$)	$L2$ es la lista de diferencias que resulta de rotar un lugar a la izquierda, en forma circular, los elementos de la lista de diferencias $L1$. Ejemplo: <i>rotacion_Id</i> ($[a,b,c/X]-X, [b,c,a/Y]-Y$)
reverse_Id (L, R)	R es la lista de diferencias que representa el inverso de la lista común L .
quicksort_Id (L, S)	S es la lista de diferencias ordenada que representa la lista común L ordenada utilizando el algoritmo quicksort.

Ejercicio 2 [prueba 2013]

Escriba el siguiente predicado en prolog. La solución no podrá invocarse recursivamente más de **n** veces, siendo **n** el largo de la lista original. Se sugiere utilizar listas de diferencias.

listaCapicua($+L, ?LCap$) **LCap** es una lista capicúa con el doble de elementos de **L**, y se cumple que **L** es prefijo de **LCap**.

Ejemplos:

listaCapicua($[a,b,c], [a,b,c,c,b,a]$).
listaCapicua($[a,b], [a,b,b,a]$).
listaCapicua($[a], [a,a]$).
listaCapicua($[], []]$).

Ejercicio 3

Escriba los siguientes predicados para árboles de enteros, representados mediante estructuras incompletas:

pre_orden (A, L)	L es una lista con los elementos del árbol binario A , obtenida al recorrerlo pre-orden.
in_orden (A, L)	L es una lista con los elementos del árbol binario A , obtenida al recorrerlo in-orden.
ins_abb (A, E)	El árbol binario de búsqueda A contiene al elemento E en la posición que le corresponde según su valor.

Ejercicio 4

Implemente los siguientes predicados sobre listas de diferencias en Prolog:

a) [prueba 2020]

`largo_id(+L,?N) ← N` es el largo de los elementos de la lista de diferencias `L`, sin contar el resto variable. Por ejemplo:

`largo_id([a,b,c,d|LR]-LR,4).`
`largo_id([c,d|LR]-LR,2).`

b) [prueba 2021]

`not_member_id(+X,?L) ← X` es un elemento que no está presente en la lista de diferencias `L` utilizada con la notación `L-LR`. Por ejemplo:

`not_member_id(6,[1,2,3,4|LR]-LR). ← Devuelve "true".`
`not_member_id(4,[1,2,3,4|LR]-LR). ← Devuelve "false".`

Ejercicio 5

Utilizando DCG, defina programas Prolog para reconocer los siguientes lenguajes:

- $L = \{a^* b^* c^*\}$
- $L = \{a^n b^n / n \geq 0\}$
- $L = \{ww^R / w \in \{a,b\}^*\}$
- $L = \{a^n b^n c^n / n \geq 0\}$
- $L = \{a^n b^m c^{n+m} / n, m \geq 0\}$
- $L = \{a^p b^m c^{p*m} / p, m \geq 0\}$

Ejercicio 6

a) Escriba una gramática en Prolog, usando la notación DCG, que permita reconocer frases como:

Los osos polares comen peces.
Los osos polares viven en la Antártida.
El investigador da alimento a los osos polares.
Los investigadores estudian el comportamiento de las aves.

b) Explique cómo haría para no permitir frases como:

Las investigador estudian el comportamiento de los aves.

c) Explique cómo haría para no permitir frases como:

Los osos polares comen la Antártida.

Ejercicio 7

a) [prueba 2010] Defina una gramática DCG para el lenguaje sobre el alfabeto $\{a,b\}$ cuyas tiras son de la forma ww , $w \in \{a,b\}^*$

b) [prueba 2012] Construya una gramática DCG que reconozca el lenguaje $L = \{y, w \in (a|b)^* / x = yww^R y\}$.

Ejercicio 8

Defina programas Prolog para reconocer los siguientes lenguajes:

- Sentencias *while*, *assign*, expresiones aritméticas y lógicas de un lenguaje imperativo [suponga que la entrada es una lista de tokens]
- La sintaxis para definir gramáticas en Prolog [llamadas DCG: *Definite Clause Grammar*] usando una gramática en Prolog.

Ejercicio 9

Considere el siguiente programa Prolog:

```

indice(Patron, Lista, Indice):-
  indice(Patron, 1, Indice, Lista, _).

indice(Patron, I, I) --> Patron.

indice(Patron, I0, I) -->
  [_],
  {I1 is I0+1},
  indice(Patron, I1, I).

```

Indique su comportamiento frente a cada una de las siguientes consultas:

- ? indice([a,b,a], [x,y,a,b,a,b,a], I).
 - La respuesta es NO
 - La respuesta es I = [a,b]
 - La respuesta es I = 3 ; I = 5
 - La respuesta es I = 1 ; I = 2 ; I = 3
 - Ninguna de las anteriores
- ? indice([a,b,a], [x,y,a,b,b], I).
 - La respuesta es NO
 - La respuesta es I = [a,b,b]
 - La respuesta es I = 3
 - La respuesta es I = 1 ; I = 2 ; I = 3
 - Ninguna de las anteriores
- ? indice([], [a,b,x,a], I).
 - La respuesta es NO
 - La respuesta es I = [a,b]
 - La respuesta es I = [a,b,x,a]
 - La respuesta es I = 1 ; I = 4
 - Ninguna de las anteriores
- ? indice([X,Y,Z], [a,b,c,d], _).
 - La respuesta es NO
 - La respuesta es X = a , Y = b, Z = c ; X=b, Y=c, Z=d
 - La respuesta es X = [a,b,c,d], Y = [], Z=[]
 - La respuesta es I = 1 ; I = 2 ; I = 3 ; I = 4
 - Ninguna de las anteriores