

TRABAJO PRACTICO N° 6

PARADIGMAS DE PROGRAMACIÓN

Dr. Pablo Javier Vidal

Unidad 3

Ejercicio 1.

¿Cuáles de los siguientes pares unifica? ¿Qué resulta de la unificación? Cuando sea necesario, proporcionar las instancias necesarias para generar una unificación exitosa.

1. $\text{bing} = \text{bong}$.
2. $2 = 2$.
3. $\text{'joe'} = \text{joe}$.
4. $\text{'2'} = 2$.
5. $\text{joe} = X$.
6. $X = Y$.
7. $J = \text{joe}, J = \text{john}$.
8. $\text{pan} = \text{pan}$
9. $\text{'Pan'} = \text{pan}$
10. $\text{'pan'} = \text{pan}$
11. $\text{Pan} = \text{pan}$
12. $\text{pan} = \text{salsa}$
13. $\text{comida}(\text{pan}) = \text{pan}$
14. $\text{comida}(\text{pan}) = X$
15. $\text{likes}(X, X) = \text{likes}(\text{joe}, \text{pizza})$.
16. $\text{father}(X) = X$.
17. $\text{comida}(X) = \text{comida}(\text{pan})$
18. $\text{comida}(\text{pan}, X) = \text{comida}(Y, \text{salchicha})$
19. $\text{comida}(X) = X$
20. $\text{comida_evento}(\text{comida}(\text{pan}), \text{bebida}(\text{cerveza})) = \text{comida_evento}(X, Y)$
21. $\text{comida_evento}(\text{comida}(\text{pan}), X) = \text{comida_evento}(X, \text{bebida}(\text{cerveza}))$
22.
 - $\text{eats}(\text{fred}, \text{tomatoes})$
 - $\text{eats}(\text{Whom}, \text{What})$
23.
 - $\text{cd}(29, \text{beatles}, \text{sgt_pepper})$.
 - $\text{cd}(A, B, \text{help})$.
24.
 - $f(X, a)$

- $f(a,X)$
- 25. ▪ $likes(jane,X)$
- $likes(X,jim).$
- 26. ▪ $f(X,Y)$
- $f(P,P)$
- 27. ▪ $f(foo,L)$
- $f(A1,A1)$

Ejercicio 2.

1. ¿Cuáles de los siguientes pares unifica? Indicar como queda formada la unificación si es posible.
 - a) $[a,b,c] = [X,Y,Z]$
 - b) $[a,b,c] = [X|Y]$
 - c) $[a,b,c] = [b|T]$
 - d) $[a,b,c] = [X,Y|Z]$
 - e) $[a,b,c] = [X,Y,Z|T]$
 - f) $[a,b,c] = [a|[b|[c[[]]]]]$
 - g) $[X,Y,Z] = [coding,is,fun]$
 - h) $[cat] = [X|Y]$
 - i) $[[the|Y]|Z] = [[X,here]|[is,here]]$

Ejercicio 3.

1. Definir la relación $primero(L,X)$ que verifique si X es el primer elemento de la lista L. Obtener la respuesta a las siguientes preguntas:
 - a) $primero([a,b,c],X).$
 - b) $primero([X,b,c],a).$
 - c) $primero([X,Y],a).$
 - d) $primero(X,a).$
2. Definir la relación $resto(L1,L2)$ que verifique si L2 es la lista obtenida a partir de la lista L1 suprimiendo el primer elemento. Obtener la respuesta a las siguientes preguntas:
 - a) $resto([a,b,c],L).$
 - b) $resto([a/L],[b,c]).$
 - c) $resto(L,[b,c]).$
3. Definir la relación $construye(X,L1,L2)$ que verifique si L2 es la lista obtenida añadiéndole X a L1 como primer elemento. Obtener la respuesta a las siguientes preguntas:
 - a) $construye(a,[b,c],L).$
 - b) $construye(X,[b,c],[a,b,c]).$
 - c) $construye(a,L,[a,b,c]).$

- d) *construye*($b, L, [a, b, c]$).
- e) *construye*($b, L, [a, b, c]$).
4. Definir la relación *pertenece*(X, L) que verifique si X es un elemento de la lista L . Utilizar el programa para responder a las siguientes cuestiones:
 - a) ¿Es c un elemento de $[a, c, b, c]$?
 - b) ¿Cuáles son los elementos de $[a, b, a]$?
 - c) ¿Cuáles son los elementos comunes de $[a, b, c]$ y $[d, c, b]$?
 5. Definir la relación *concatena*($L1, L2, L3$) (equivalente a append) que verifique si $L3$ es la lista obtenida escribiendo los elementos de $L2$ a continuación de los elementos de $L1$. Utilizar el programa para responder a las siguientes cuestiones:
 - a) ¿Qué lista hay que añadirle al lista $[a, b]$ para obtener $[a, b, d]$?
 - b) ¿Qué listas hay que concatenar para obtener $[a, b]$?
 - c) ¿Cuál es el último elemento de $[b, a, d]$?
 6. Un palíndromo es una palabra que se lee igual en los dos sentidos, por ejemplo “oso”. Definir la relación *palíndromo*(L) que verifique si la lista L es un palíndromo.
 7. Definir la relación *último*(X, L) (equivalente a last) que verifique si X es el último elemento de la lista L . (versión append, versión reverse, versión recursiva)
 8. Utilizando el predicado select, definir la relación *inserta*($X, L1, L2$) que verifique si $L2$ es una lista obtenida insertando X en $L1$. Compruebe el resultado de la consulta *inserta*($a, [1, 2], L$).
 9. Utilizando el predicado append, definir la relación *sublista*($L1, L2$) que verifique si $L1$ es una sublista de $L2$.
 10. Definir la relación *subconjunto*($L1, L2$) que verifique si $L2$ es un subconjunto de $L1$.
 11. Definir la relación *máximo*(X, Y, Z) (equivalente a max) que verifique si Z es el máximo de X e Y .
 12. Definir la relación *mcd*(X, Y, Z) que verifique si Z es el máximo común divisor de X e Y .
 13. Definir la relación *longitud*(L, N) que se verifique si N es la longitud de la lista L .
 14. Una lista está acotada si todos sus elementos son menores que su longitud. Definir la relación *lista_acotada*(L) que verifique si todos los elementos de la lista de números L son menores que la longitud de L . (usar length/2)
 15. Definir la relación *max_lista*(L, X) que se verifique si X es el máximo de la lista de números L .
 16. Definir la relación *suma_lista*(L, X) (equivalente a sumlist) que se verifique si X es la suma de los elementos de la lista de números L .
 17. Definir la relación *ordenada*(L) que se verifique si la lista de números L está ordenada de manera creciente.
 18. Definir la relación *lista*(N, L) que verifique si L es la lista de longitud N cuyos elementos son N .
 19. Definir la relación *lista_de_números*(N, M, L) (equivalente a numlist) que verifique si L es la lista de los números desde N hasta M , ambos inclusive.
 20. Definir la relación *entre*($N1, N2, X$) (equivalente a between) que se verifique si X es un número entero tal que $N1 \leq X \leq N2$.

Ejercicio 4.

1. Define un predicado $\text{multirot}(Xs, MRXs)$ que, dadas las listas Xs y $MRXs$, se satisfice cuando $MRXs$ es la concatenación de varias rotaciones (al menos una) de Xs . Por ejemplo:

?- $\text{multirot}([a,b,c],[a,b,c,c,a,b,b,c,a,c,a,b,a,b,c])$.

Yes

?- $\text{multirot}([a,b,c],[a,b,c,a,c])$.

No

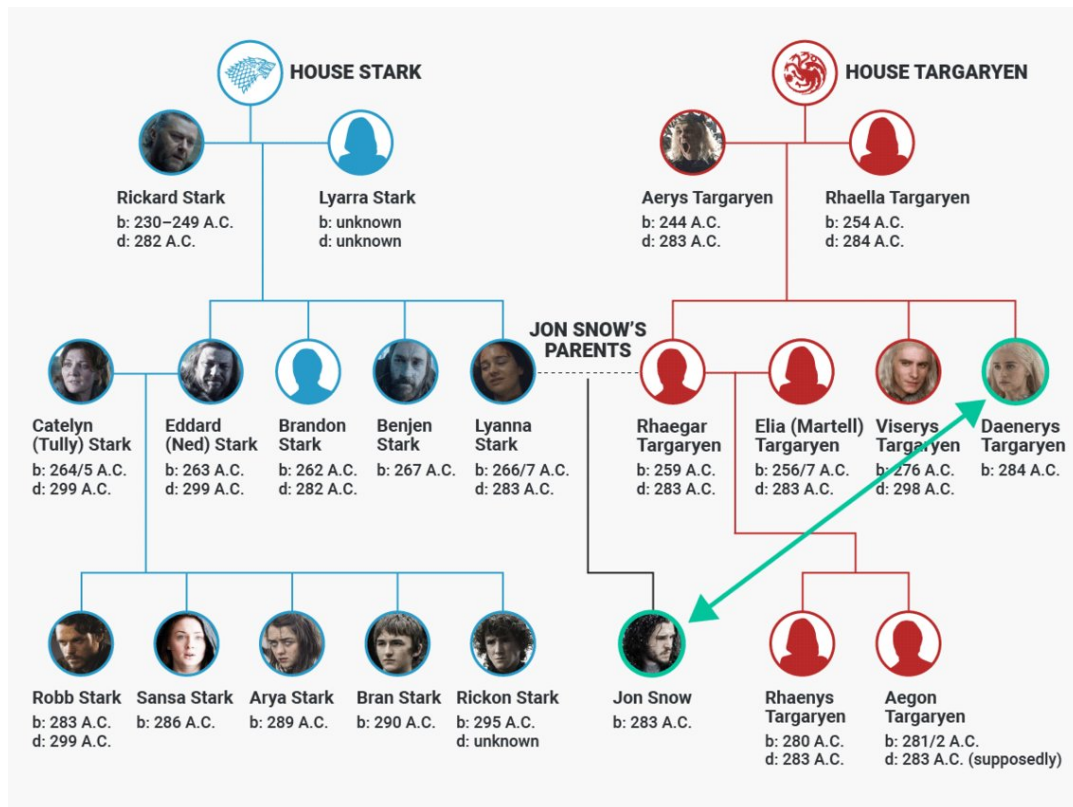
2. Define un predicado $\text{son_consecutivas}(N, Xs)$ que se satisfaga cuando Xs sea una lista donde cada número i entre 1 y N aparece i veces consecutivas. Por ejemplo:

?- $\text{son_consecutivas}(5, [3,3,3,1,2,2,5,5,5,5,5,4,4,4,4])$.

Yes

Ejercicio 5.

Dado el siguiente gráfico que muestra los arboles genealógicos de la familia Stark y Targaryen en la serie Games of Thrones:



1. Definir los siguientes hechos:

- pareja/2
- amante/2
- muerto/1
- revivido/2, revivido por el rey de la noche
- bastardo/1
- persona/3 con funtores: nombre y apellido, fecha de nacimiento,
- padres/3, H es hijo del P y M

- casa/1
- pertenece a casa/2.

2. Definir las siguientes reglas

- padre/2
- madre/2
- hermano/2
- lista_hermanos/2
- lista_casa/2
- es_mas_viejo
- revividosPorElRey/2
- hijos/3
- descendiente/2
- bastardos_de_casa/2
- hijos_no_bastardos/3
- lista_hermanos_ordenados_por_mas_viejo/2.
- hermanoMasViejo/2
- quienes forman parte de la casa pero no tienen sangre de esa casa.
- quienes tienen mascotas
- que mascotas tiene cada uno.

Ejercicio 6.

1. Compare el paradigma imperativo versus el paradigma declarativo
2. Analice porqué, en la programación lógica, los bucles son modelados a través de la recursividad.
3. Enumere las características de Prolog que lo alejan del paradigma lógico.
4. En que momento el concepto de Unificación puede ser utilizado
5. ¿Qué es la reflexión en un lenguaje de programación? Explique qué mecanismos provee prolog para realizar reflexión. ¿Conoce algún otro lenguaje que provea mecanismos para estos conceptos? Justifique y compárelos con los mecanismos provistos por Prolog.
6. Explique cómo Prolog implementa la aritmética simple. En particular explique el uso del predicado is y sus principales características.
7. Como funciona el concepto de Backtracking
8. Considere los siguientes predicados Prolog:

- a) suma1(X,Y,Z):- Z is X+Y.
- b) suma2(X,Y,Z):- Z=X+Y.

¿Por qué no son semánticamente equivalentes? ¿Existe algún caso a donde se comporten de la misma manera?