Tarea 2: Prolog

Carlo Miguel Herrera Di Giacinto (18-10451)

Parte I. Implementación

1. Números de Church.

Considere la siguiente definición de los números de Church:

> "El cero (0) es representado por la constante universal ***I*** luego cada número natural se representa como la aplicación repetida de una función ***up*** sobre el cero. El número es entonces igual a la cantidad de aplicaciones de dicha función."

Por ejemplo:

zero → 0

next(zero) → 1

next(next(zero)) → 2

next(next(next(zero))) → 3

Tomando en cuenta la definición anterior, se desea que implemente en Prolog una calculadora para números de Church. De los tres argumentos, los primeros dos deben estar siempre instanciados. El tercero puede o no estarlo

> *(**Nota:** No debe transformar los números de Church en números enteros).*

a) Implemente el predicado `suma/3`, que debe ser cierto cuando el tercer argumento corresponda a la suma de los primeros dos argumentos.

```
suma(zero, Y, Y).
suma(next(X), Y, next(Z)) :-
    suma(X, Y, Z).
```

b) Implemente el predicado `resta/3`, que debe ser cierto cuando el tercer argumento corresponda a la resta positiva de los primeros dos argumentos (nótese que la resta está indefinida si el primer argumento es menor que el segundo).

```
resta(X, zero, X).
resta(next(X), next(Y), Z) :-
    resta(X, Y, Z).
```

c) Implemente el predicados `producto/3`, que debe ser cierto cuando el tercer argumento corresponda al producto de los primeros dos argumentos.

```
producto(zero, _, zero).
producto(next(X), Y, Z) :-
    producto(X, Y, W),
    suma(Y, W, Z).
```

2. Árboles.

Considere hechos de la forma `arco(A, B)` en ProLog, representando una conexión dirigida desde un nodo A hasta un nodo B.

a) Implemente un predicado `hermano(A, B)` que se satisfaga si existe nodo C, tal que exista un arco desde C hasta A y un arco desde C hasta B.

```
hermano(A,B) :- arco(A,C), arco(C,B).
```

b) Implemente un predicado `alcanzable(A, B)` que se satisfaga si existe algún camino, siguiendo 0 o más arcos desde A hasta B. *(Puede suponer que el grafo que inducen los arcos es acíclico).*

```
alcanzable(A,A).
alcanzable(A,B) :- arco(A,C), alcanzable(C,B).
```

c) Implemente un predicado `lca(A, B, C)` que se satisfaga si C alcanza tanto A como B; y no existe ningún C' alcanzable desde C, que a su vez alcance también a A y B. *(Puede suponer que el grafo que inducen los arcos es acíclico).*

```
lca(A,A,A) :- !.
lca(A,B,C) :- arco(C,A), arco(C,B), !.
lca(A,B,C) :-
    alcanzable(C,A),
    alcanzable(C,B),
    not(alcanzable(C,Z)), C \= Z, alcanzable(Z,A), alcanzable(Z,B).
```

d) Implemente un predicado `tree(A)` que se satisfaga si los nodos alcanzables desde A (en conjunto con los arcos en cuestión) forman un árbol.

```

```

Parte II. Investigación