**Estructuras de Datos**

**Uso de la clausula cut**



* Sirve para modificar el proceso de backtracking, cortando ramas del arbol de busqueda de solcuones que ya no se necesitan recorrer.
* Ayuda en la eficiencia de ejecucion de un programa en Prolog.
* Util cuando basta una sola solucion, o cuando hay riesgo de caer en una busqueda infinita de soluciones.

**El cut corta la ejecucion de los terminos que aparecen antes que el.**

**Formato**

* Solo se utiliza el simbolo ! en el cuerpo de una regla.
* Es una clausula que siempre es verdadera por default.
* Al ejecutarse, provoca que en el backtracking, ya no se busquen alternativas de éxito para los terminos anteriores del cut y para la cabeza de la clausula.

**Listas en Prolog**

* [X|Y] X-head, Y-tail
* [1,2, 3, 4] 🡪 head: 1 y tail: [2,3,4]
* [1|[2,3,4]]

**Funciones Predefinidas (Pensamiento Logico)**

1. member(X, Y)
   1. Verdadero cuando X es elemento de Y.
   2. Si no te dan el valor de X, te da el valor de todos (;)
2. length(X, Y)
   1. Verdadero cuando Y es el numero de elementos de la lista.
3. append(X, Y, Z)
   1. Verdadero cuando Z es la concatenacion de X e Y.

**Pensamiento Recursivo**

* Caso Base: cuando la lista esta vacia o en el numero cero.
* Caso General:
  + Sobre el numero anterior a n (n – 1).
  + Sobre la lista [x|y] pero sin su primer elemento.

**Funciones**

Las funciones son tipos especiales de relaciones, es una forma de hacer referencia a objetos.

Se pueden usar como argumentos.

Para otras estructuras de datos se puede utilizar la representacion por medio de funciones.

Ejemplo

arbol(root, subIzq, subDer)



Ejemplo

% Recorridos

preorden(arbol(R, I, D), L):-

!, preorden(I, Rizq),



preorden(D, Rder),



append([R|Rizq], Rder, L).



preorden(nil, []).

