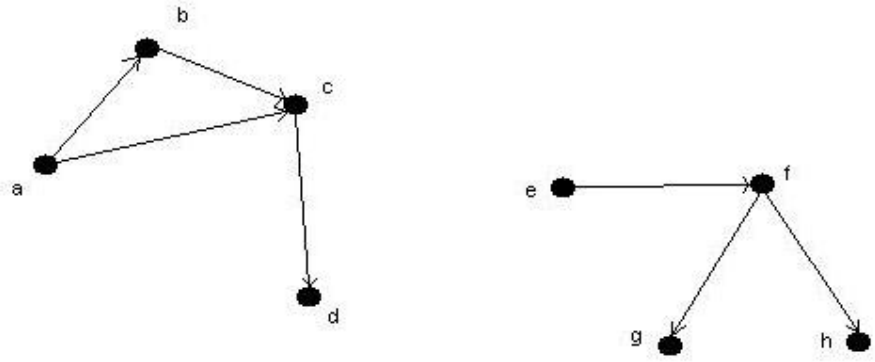
Lógico 5: Recursividad, con y sin listas

Ejercicio 1

Escribir un programa Prolog que describa un grafo dirigido, de forma tal que pueda preguntar si hay camino desde un nodo hasta otro.
P.ej. en este grafo



sí hay camino desde *a* hasta *d*, pero no hay camino ni desde *d* hasta *a* ni desde *a* hasta *f*.
No tener en cuenta ciclos.

Verificar que el predicado definido para la consulta es inversible para todos sus argumentos.

Ejercicio 2

Escribir un programa Prolog al que le pueda pedir el valor de la serie de Fibonacci para cualquier natural.
La serie de Fibonacci se define así:

- fib(1) = 1.
- fib(2) = 1.
- fib(n) = fib(n-1) + fib(n-2), si n > 2.

Ejercicio 3

Armar un programa para una central telefónica que permita realizar derivaciones de llamadas. La central funciona de la siguiente manera:

En la base de conocimientos tenemos registrados a todos los empleados y sus jefes, y el área al que pertencen los jefes. Por ejemplo:

```
empleado(juan).
empleado(marcelo).
empleado(adriana).
empleado(francisco).
empleado(alberto).
empleado(cristian).
empleado(mariana).
jefe(marcelo, juan).
jefe(marcelo, adriana).
jefe(mariana, cristian).
area(marcelo, administracion)
area(mariana, ventas).
```

En la base se registran también los internos de las personas. Por ejemplo:

```
interno(marcelo, 2244).
interno(adriana, 2245).
interno(francisco, 4441).
interno(alberto, 4442).
interno(mariana, 1212).
```

También se registran los gerentes y sus asistentes.

```
gerente(alberto, administracion).
asistente(francisco, alberto).
```

Programar las siguientes reglas:

1- internoDe(Persona, Interno)

Relaciona una persona con su interno. Si la persona no tiene interno, relaciona una persona con el interno de su jefe.

2- quienAtiende(Interno, Persona)

Dice quien atiende un interno. Si llaman al interno de un jefe, hace que atienda cualquiera de sus empleados. Si llaman al interno de un gerente, hace que atienda su asistente. En cualquier otro caso, atiende la persona dueña del interno.

3- dependeDe(Persona, Persona)

Dice si la primer persona depende de la segunda persona. Una persona "A" depende de una persona "B", si "B" es jefe de "A", o si "B" es gerente del área donde trabaja "A".

4- puedeTransferir(interno, interno)

Dos internos se pueden transferir si pertenecen a empleados que dependen de la misma persona. Para resolverla utilizar la regla dependeDe.

5- pertenecen ([internos], [internos]).
"Quita" de una lista de números telefónicos, los que NO pertenecen a internos de la empresa.

6- personas([Personas], [internos])
"Devuelve" de una lista de personas, sus internos. Para resolverla utilizar la regla internoDe.

Ejercicio 4, Recursividad con listas

Definir los siguientes predicados:

a. suma(L,N): relaciona una lista de números L con la suma. La suma de una lista vacía se define como 0.

b. encolar(E,L,LconE): relaciona un elemento E, una lista L, y la lista que resulta de agregar E al final de L.
P.ej.

- ?- encolar([1,3,5],7,[1,3,5,7]). devuelve Yes
- ?- encolar([1,3,5],7,[1,3,7]). devuelve No
- ?- encolar([1,3,5],7,[1,3,5,8]). devuelve No

c. maximo(L,N): relaciona una lista de números L con su elemento más grande. La lista vacía no tiene máximo.

d. elementoEn(L,P,E): relaciona una lista L, un número P y el elemento que está en la posición P de L.
El primer elemento es el 1.
P.ej.

- ?- elementoEn([2,5,8,11,14],3,8). devuelve Yes
- ?- elementoEn([2,5,8,11,14],3,X). devuelve 8
- ?- elementoEn([2,5,8,11,14],7,X). devuelve No (porque la lista tiene menos de 7 elementos)