

Ejercicios Integradores

Saltos

En una competencia de saltos, cada competidor puede hacer hasta 5 saltos; a cada salto se le asigna un puntaje de 0 a 10. Se define el predicado `puntajes` que relaciona cada competidor con los puntajes de sus saltos, p.ej.

```
puntajes(hernan, [3, 5, 8, 6, 9]).  
puntajes(julio, [9, 7, 3, 9, 10, 2]).  
puntajes(ruben, [3, 5, 3, 8, 3]).  
puntajes(roque, [7, 10, 10]).
```

Se pide armar un programa Prolog que a partir de esta información permita consultar:

1. Qué puntaje obtuvo un competidor en un salto, p.ej. qué puntaje obtuvo Hernán en el salto 4 (respuesta: 6).
2. Si un competidor está descalificado o no. Un competidor está descalificado si hizo más de 5 saltos. En el ejemplo, Julio está descalificado.
3. Si un competidor clasifica a la final. Un competidor clasifica a la final si la suma de sus puntajes es mayor o igual a 28, o si tiene dos saltos de 8 o más puntos.

Ayuda: investigar predicado `nth1/3` y `nth0/3` en el prolog.

Subtes

En este ejercicio viene bien usar el predicado `nth1/3`, que relaciona un número, una lista, y el elemento de la lista en la posición indicada por el número (empezando a contar de 1). P.ej. prueben estas consultas

```
?- nth1(X,[a,b,c,d,e],d).
?- nth1(4,[a,b,c,d,e],X).
?- nth1(Orden,[a,b,c,d,e],Elem).
?- nth1(X,[a,b,c,d,e],j).
?- nth1(22,[a,b,c,d,e],X).
```

Tenemos un modelo de la red de subtes, por medio de un predicado `linea` que relaciona el nombre de la línea con la lista de sus estaciones, en orden. P.ej. (reduciendo las líneas)

```
linea(a,[plazaMayo,peru,lima,congreso,miserere,rioJaneiro,primeraJunta,nazca]).
linea(b,[alem,pellegrini,callao,pueyrredonB,gardel,medrano,malabia,lacroze,losIncas,urquiza]).
linea(c,[retiro,diagNorte,avMayo,independenciaC,plazaC]).
linea(d,[catedral,nueveJulio,medicina,pueyrredonD,plazaItalia,carranza,congresoTucuman]).
linea(e,[bolivar,independenciaE,pichincha,jujuy,boedo,varela,virreyes]).
linea(h,[lasHeras,santaFe,corrientes,once,venezuela,humberto1ro,inclan,caseros]).
combinacion([lima,avMayo]).
combinacion([once,miserere]).
combinacion([pellegrini,diagNorte,nueveJulio]).
combinacion([independenciaC,independenciaE]).
combinacion([jujuy,humberto1ro]).
combinacion([santaFe,pueyrredonD]).
combinacion([corrientes,pueyrredonB]).
```

No hay dos estaciones con el mismo nombre.

Se pide armar un programa Prolog que a partir de esta información permita consultar:

1. En qué línea está una estación, predicado **estaEn/2**.
2. dadas dos estaciones de la misma línea, cuántas estaciones hay entre ellas, p.ej. entre Perú y Primera Junta hay 5 estaciones. Predicado **distancia/3** (relaciona las dos estaciones y la distancia).
3. Dadas dos estaciones de distintas líneas, si están a la misma altura (o sea, las dos terceras, las dos quintas, etc.), p.ej. Independencia C y Jujuy que están las dos cuartas. Predicado **mismaAltura/2**.
4. Dadas dos estaciones, si puedo llegar fácil de una a la otra, esto es, si están en la misma línea, o bien puedo llegar con una sola combinación. Predicado **viajeFacil/2**.

Viajes

Una agencia de viajes lleva un registro con todos los vuelos que maneja de la siguiente manera:

`vuelo(Codigo de vuelo, capacidad en toneladas, [lista de destinos])`.

Esta lista de destinos está compuesta de la siguiente manera:

`escala(ciudad, tiempo de espera)`

`tramo(tiempo en vuelo)`

Siempre se comienza de una ciudad (escala) y se termina en otra (no puede terminar en el aire al vuelo), con tiempo de vuelo entre medio de las ciudades. Considerar que los viajes son de ida y de vuelta por la misma ruta.

`vuelo(ARG845, 30, [escala(rosario,0), tramo(2), escala(buenosAires,0)])`.

`vuelo(MH101, 95, [escala(kualaLumpur,0), tramo(9), escala(capeTown,2), tramo(15), escala(buenosAires,0)])`.

`vuelo(DLH470, 60, [escala(berlin,0), tramo(9), escala(washington,2), tramo(2), escala(nuevaYork,0)])`.

`vuelo(AAL1803, 250, [escala(nuevaYork,0), tramo(1), escala(washington,2), tramo(3), escala(ottawa,3), tramo(15), escala(londres,4), tramo(1), escala(paris,0)])`.

`vuelo(BLE849, 175, [escala(paris,0), tramo(2), escala(berlin,1), tramo(3), escala(kiev,2), tramo(2), escala(moscu,4), tramo(5), escala(seul,2), tramo(3), escala(tokyo,0)])`.

`vuelo(NPO556, 150, [escala(kiev,0), tramo(1), escala(moscu,3), tramo(5), escala(nuevaDelhi,6), tramo(2), escala(hongKong,4), tramo(2), escala(shanghai,5), tramo(3), escala(tokyo,0)])`.

`vuelo(DSM3450, 75, [escala(santiagoDeChile,0), tramo(1), escala(buenosAires,2), tramo(7), escala(washington,4), tramo(15), escala(berlin,3), tramo(15), escala(tokyo,0)])`.

Definir los siguientes predicados; en todos vamos a identificar cada vuelo por su código.

1. **tiempoTotalVuelo/2** : Relaciona un vuelo con el tiempo que lleva en total, contando las esperas en las escalas (y eventualmente en el origen y/o destino) más el tiempo de vuelo.
2. **escalaAburrida/2** : Relaciona un vuelo con cada una de sus escalas aburridas; una escala es aburrida si hay que esperar mas de 3 horas.
3. **ciudadesAburridas/2** : Relaciona un vuelo con la lista de ciudades de sus escalas aburridas.
4. **vueloLargo/1**: Si un vuelo pasa 10 o más horas en el aire, entonces es un vuelo largo. OJO que dice "en el aire", en este punto no hay que contar las esperas en tierra. **conectados/2**: Relaciona 2 vuelos si tienen al menos una ciudad en común.
5. **bandaDeTres/3**: Relaciona 3 vuelos si están conectados, el primero con el segundo, y el segundo con el tercero.
6. **distanciaEnEscalas/3**: Relaciona dos ciudades que son escalas del mismo vuelo y la cantidad de escalas entre las mismas; si no hay escalas intermedias la distancia es 1. P.ej. París y Berlín están a distancia 1 (por el vuelo BLE849), Berlín y Seúl están a distancia 3 (por el mismo vuelo). No importa de qué vuelo, lo que tiene que pasar es que haya algún vuelo que tenga como escalas a ambas ciudades. Consejo: usar `nth1`.
7. **vueloLento/1**: Un vuelo es lento si no es largo, y además cada escala es aburrida.

Cosas y Países

Se organiza un juego que consiste en ir buscando distintos objetos por el mundo. Cada participante está en un determinado nivel, cada nivel implica ciertas tareas, cada tarea consiste en buscar un objeto en una ciudad.

Representamos las tareas como funtores **buscar**(Cosa,Ciudad), y definimos el predicado **tarea/2** de esta forma:

```
tarea(basico, buscar(libro, jartum)).
tarea(basico, buscar(arbol, patras)).
tarea(basico, buscar(roca, telaviv)).
tarea(intermedio, buscar(arbol, sofia)).
tarea(intermedio, buscar(arbol, bucarest)).
tarea(avanzado, buscar(perro, bari)).
tarea(avanzado, buscar(flora, belgrado)).
```

O sea, si estoy en el nivel básico, mis tareas posibles son buscar un libro en Jartum, un árbol en Patras o una roca en Tel Aviv.

Para definir en qué nivel está cada participante se define el predicado **nivelActual/2**, de esta forma:

```
nivelActual(pepe, basico).
nivelActual(lucy, intermedio).
nivelActual(juancho, avanzado).
```

También vamos a necesitar saber qué idioma se habla en cada ciudad, qué idiomas habla cada persona, y el capital actual de cada persona. Esto lo representamos con los predicados **idioma/2**, **habla/2** y **capital/2**:

```
idioma(alejandria, arabe).
idioma(jartum, arabe).
idioma(patras, griego).
idioma(telaviv, hebreo).
idioma(sofia, bulgaro).
idioma(bari, italiano).
idioma(bucarest, rumano).
idioma(belgrado, serbio).
```

```
habla(pepe, bulgaro).
habla(pepe, griego).
habla(pepe, italiano).
habla(juancho, arabe).
habla(juancho, griego).
habla(juancho, hebreo).
habla(lucy, griego).
```

```
capital(pepe, 1200).
capital(lucy, 3000).
capital(juancho, 500).
```

Definir los siguientes predicados:

1. **destinoPosible/2** e **idiomaUtil/2**.

destinoPosible/2 relaciona personas con ciudades; una ciudad es destino posible para un nivel si alguna tarea que tiene que hacer la persona (dado su nivel) se lleva a cabo en la ciudad. P.ej. los destinos posibles para Pepe son: Jartum, Patras y Tel Aviv.

idiomaUtil/2 relaciona niveles con idiomas: un idioma es útil para un nivel si en alguno de los

destinos posibles para el nivel se habla el idioma. P.ej. los idiomas útiles para Pepe son: árabe, griego y hebreo.

2. **excelenteCompañero/2**, que relaciona dos participantes. P2 es un excelente compañero para P1 si habla los idiomas de todos los destinos posibles del nivel donde está P1. P.ej. Juancho es un excelente compañero para Pepe, porque habla todos los idiomas de los destinos posibles para el nivel de Pepe.

Asegurar que el predicado sea inversible para los dos parámetros.

3. **interesante/1**: un nivel es interesante si se cumple alguna de estas condiciones
 - a. todas las cosas posibles para buscar en ese nivel están vivas (las cosas vivas en el ejemplo son: árbol, perro y flor)
 - b. en alguno de los destinos posibles para el nivel se habla italiano.
 - c. la suma del capital de los participantes de ese nivel es mayor a 10000

Asegurar que el predicado sea inversible.

4. **complicado/1**: un participante está complicado si: no habla ninguno de los idiomas de los destinos posibles para su nivel actual; está en un nivel distinto de básico y su capital es menor a 1500, o está en el nivel básico y su capital es menor a 500.
5. **homogeneo/1**: un nivel es homogéneo si en todas las opciones la cosa a buscar es la misma. En el ejemplo, el nivel intermedio es homogéneo, porque en las dos opciones el objeto a buscar es un árbol.

Asegurar que el predicado sea inversible.

Ayuda: hay que saber de alguna forma si una lista tiene un único elemento, p.ej. la lista [3,3,3,3] tiene un único elemento (el 3) mientras que la lista [2,1,2,4] no. Tal vez convenga definir un predicado aparte para este problema.

6. **políglota/1**: una persona es políglota si habla al menos tres idiomas. En general: es válido agregar los predicados necesarios para poder garantizar inversibilidad o auxiliares para resolver cada ítem, y usar en un ítem los predicados definidos para resolver ítems anteriores.

TEG!

En vista de que gran parte de las personas (chicos y grandes) abandonan los juegos clásicos por modernos juegos de PC, la juguetería de Lanús Quindimil SRL, decide llevar su negocio al terreno digital para poder competir con las nuevas formas de esparcimiento. Así es como se comunican con nosotros solicitándonos que realicemos el desarrollo de ciertas consultas para un tablero de TEG, que se realizan cada cierto tiempo. Cada vez que se realizan las consultas, vamos a contar con ciertos hechos como los que ejemplificamos a continuación.

```
/* distintos paises */
paisContinente(argentina, americaDelSur).
paisContinente(bolivia, americaDelSur).
paisContinente(brasil, americaDelSur).
paisContinente(chile, americaDelSur).
paisContinente(ecuador, americaDelSur).
paisContinente(alemania, europa).
paisContinente(españa, europa).
paisContinente(francia, europa).
paisContinente(inglaterra, europa).
paisContinente(aral, asia).
paisContinente(china, asia).
paisContinente(gobi, asia).
paisContinente(india, asia).
paisContinente(iran, asia).
```

```
/*países importantes*/
paisImportante(argentina).
paisImportante(kamchatka).
paisImportante(alemania).
```

```
/*países limítrofes*/
limitrofes([argentina,brasil]).
limitrofes([bolivia,brasil]).
limitrofes([bolivia,argentina]).
limitrofes([argentina,chile]).
limitrofes([españa,francia]).
limitrofes([alemania,francia]).
limitrofes([nepal,india]).
limitrofes([china,india]).
limitrofes([nepal,china]).
limitrofes([afganistan,china]).
limitrofes([iran,afganistan]).
```

```
/*distribución en el tablero */
ocupa(argentina, azul, 4).
ocupa(bolivia, rojo, 1).
ocupa(brasil, verde, 4).
ocupa(chile, negro, 3).
ocupa(ecuador, rojo, 2).
ocupa(alemania, azul, 3).
```

```

ocupa(españa, azul, 1).
ocupa(francia, azul, 1).
ocupa(inglaterra, azul, 2).
ocupa(aral, negro, 2).
ocupa(china, verde, 1).
ocupa(gobi, verde, 2).
ocupa(india, rojo, 3).
ocupa(iran, verde, 1).

/*continentes*/
continente(americaDelSur).
continente(europa).
continente(asia).

/*objetivos*/
objetivo(rojo, ocuparContinente(asia)).
objetivo(azul, ocuparPaíses([argentina, bolivia, francia, inglaterra, china])).
objetivo(verde, destruirJugador(rojo)).
objetivo(negro, ocuparContinente(europa)).

```

Se solicita construir un programa que permita la resolución de las siguientes consultas.

Todos los predicados deben ser inversibles, salvo aclaración explícita en contrario.

1. **estaEnContinente/2**: Relaciona un jugador y un continente si el jugador ocupa al menos un país en el continente.
2. **cantidadPaíses/2**: Relaciona a un jugador con la cantidad de países que ocupa.
3. **ocupaContinente/2**: Relaciona un jugador y un continente si el jugador ocupa totalmente al continente.
4. **leFaltaMucho/2**: Relaciona a un jugador y un continente si al jugador le falta ocupar más de 2 países de dicho continente.
5. **sonLimitrofes/2**: Relaciona 2 países si son limítrofes.
6. **esGroso/1**: Un jugador es groso si cumple alguna de estas condiciones:
 - a. ocupa todos los países importantes,
 - b. ocupa más de 10 países
 - c. o tiene más de 50 ejércitos.
7. **estaEnElHorno/1**: un país está en el horno si todos sus países limítrofes están ocupados por el mismo jugador que no es el mismo que ocupa ese país.
8. **esCaotico/1**: un continente es caótico si hay más de tres jugadores en él.
9. **capoCannoniere/1**: es el jugador que tiene ocupado más países.
10. **ganadooor/1**: un jugador es ganador si logro su objetivo