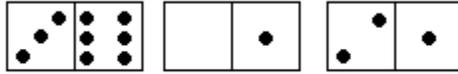


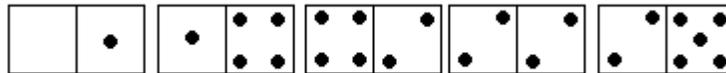
LOGICO – DOMINÓ LINEAL

El juego usual del dominó tiene 28 fichas diferentes. Cada ficha es rectangular y tiene grabado en cada extremo un número de puntos entre 0 y 6. P.ej., estas son algunas fichas:

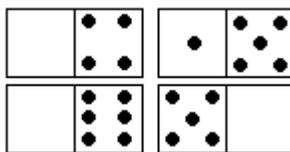


todas las fichas son distintas entre sí.

Siguiendo las reglas de este juego en particular, las fichas se colocan formando una cadena de tal manera que cada par de fichas consecutivas tienen iguales números correspondientes a los dos extremos que se tocan. El siguiente ejemplo podría ser el estado del juego en un momento



Cada jugador posee un conjunto de piezas, las cuales utiliza por turno para ir agregando a la cadena de piezas del juego; puede agregar su ficha en cualquiera de los dos extremos. P.ej. en el ejemplo anterior las siguientes fichas se pueden agregar:



(0,4) se puede agregar por la izquierda;
(0,6) también por la izquierda;
(5,1) por la derecha;
(5,0) por cualquier extremo

Cuando un jugador no puede agregar ninguna pieza, este debe ceder su turno.

Lo que sigue es parte de la base de conocimiento

```
tieneFicha(carlos, ficha(0,4)).
tieneFicha(carlos, ficha(0,6)).
tieneFicha(carlos, ficha(5,1)).
tieneFicha(german, ficha(5,0)).
tieneFicha(miguel, ficha(3,2)).
tieneFicha(miguel, ficha(3,3)).
tieneFicha(juan, ficha(1,6)).

estado([ficha(0,1), ficha(1,4), ficha(4,2), ficha(2,2), ficha(2,5)]).
```

Se pide lo siguiente

1. Relacionar una persona con una ficha que tiene. La ficha puede estar al derecho o al revés. En el ejemplo:

```
?- tiene(X, ficha(4,0)).
X = carlos;
No.
?- tiene(juan, F).
F = ficha(1,6);
F = ficha(6,1);
No.
```

2. Armar el predicado puedeSeguir/2 que relaciona a dos personas, de manera tal que si una persona jugara cualquiera de sus fichas, la otra también puede hacerlo. Por ejemplo:

```
?- puedeSeguir(german,X) .  
X= carlos;  
X= carlos;  
X= carlos;  
No.
```

carlos es la única persona que puede jugar luego que juegue german, ya que tiene 2 fichas con cero y una con un cinco (siguiendo a la única ficha (5,0) de germán).

3. Armar el predicado seguidores/2, que funciona como el anterior pero la relación se establece entre una persona y la lista de personas que pueden jugar luego. En particular, la lista no debe tener repetidos.
4. Revisar los predicados anteriores y modificarlos de manera que queden inversibles, de ser necesario.
5. Resolver el predicado cedeTurno/1, que se verifica para los jugadores que deben ceder su turno. En el ejemplo:

```
?-cedeTurno(X) .  
X = miguel;  
X = juan;  
No.
```
6. Resolver el predicado astuto/1, que se verifica para los jugadores que pueden jugar una ficha de manera que nadie más pueda jugar posteriormente (no importa si el mismo jugador tampoco puede jugar).