

Hoja 3**Tema 3.2: Representación del conocimiento. Lógica y Prolog**

1) Dado un conjunto de bloques sobre una mesa, representados en la figura por letras, queremos representar como hechos Prolog las relaciones entre los bloques. Para ello, utilizaremos dos predicados: *sobre*(X, Y) para indicar que el bloque X está sobre el bloque Y, e *izquierda*(X, Y) para indicar que el bloque X está inmediatamente a la izquierda del bloque Y, estando ambos bloques en la base de las pilas.

D		
C	G	
B	F	I
A	E	H

Definir un nuevo predicado *por_encima_de*(X, Y) que se cumpla cuando el bloque X esté en la misma pila que el bloque Y, en cualquier posición superior. Definir otro predicado *pila_izquierda*(X, Y) que se cumpla cuando el bloque X esté en la pila situada inmediatamente a la izquierda de la pila en la que está bloque Y. A continuación, construir las siguientes preguntas:

- ¿Qué bloques están por encima de otros bloques?
- ¿Qué bloques están por encima del bloque F o en una pila situada inmediatamente a la izquierda de la pila en la que está F?
- ¿Qué bloques están por encima de otros bloques pero no están en una pila situada inmediatamente a la izquierda de algún bloque?

2) Queremos definir predicados para razonar sobre árboles genealógicos, suponiendo que las relaciones familiares están definidas con un único predicado de aridad 4. El predicado se denomina *descendiente_directo* y corresponde a la relación hijo/a. Los parámetros (en orden) son: nombre del padre, nombre de la madre, nombre del hijo/a y sexo (varón o mujer).

Definir los siguientes predicados exclusivamente en función de *descendiente_directo*/4:

<i>padre</i> /2	<i>madre</i> /2	<i>hijo</i> /2	<i>abuelo</i> /2
<i>hermana</i> /2	<i>tío</i> /2	<i>prima</i> /2	<i>ascendiente</i> /2

3) Se dispone de una base de hechos en Prolog sobre relaciones familiares establecidos mediante los siguientes predicados:

descendiente_directo(Padre, Madre, Hij@, Sexo)
casados(Cónyuge1, Cónyuge2)

Considerando que la relación *progenitor*/2 corresponde a la relación padre o madre, se pide definir en Prolog este predicado, así como los predicados *hermanastro*/2 y *familiar_político*/2 en función de alguno o de los dos predicados anteriores.

Nota: Se considera que una persona es *hermanastro* de P si es varón y no tiene ningún progenitor común con P, pero uno de sus progenitores está casado con un progenitor de P. Si Q es el cónyuge de P, consideraremos *familiar político* de P a cualquier persona que sea ascendiente de Q o que sea hijo de cualquier ascendiente de Q, sin ser directamente ascendiente de Q ni coincidir con Q. Es aconsejable definir el predicado auxiliar *ascendiente*/2 en términos de *progenitor*/2 para definir *familiar_político*/2.

Hoja 3

Tema 3.2: Representación del conocimiento. Lógica y Prolog

4) Dado un mapa que representa la situación de distintas ciudades, se quiere construir un sistema de representación en Prolog que permita contestar preguntas sobre la posición relativa de dos ciudades, con las siguientes características:

- Se introducirán exclusivamente hechos correspondientes a las relaciones “estar al norte de” y “estar al oeste de” y sólo entre las ciudades más próximas entre sí. Por ejemplo, si suponemos 9 ciudades distribuidas en una cuadrícula:

A	B	C
D	E	F
G	H	I

sólo se establecerán como hechos: “A está al norte de D”, “A está al oeste de B”, etc.

- El sistema de representación será capaz de inferir todas las relaciones inversas de las dadas directamente, es decir, las relaciones “estar al sur de” y “estar al este de”.
- Se inferirán nuevas relaciones por transitividad. Por ejemplo, sabiendo que “A está al norte de D” y que “D está al norte de G” se inferirá que “A está al norte de G”.
- Se inferirán las relaciones noroeste, noreste, suroeste y sureste a partir de los hechos iniciales. Por ejemplo, se podrá inferir que “C está al noreste de G”.

El predicado que se utilizará para consultar al sistema será *situacion(X, Y, Z)*, que significa que la situación relativa de la ciudad X con respecto a la ciudad Y es Z, es decir que “X está al Z de Y”. Por ejemplo:

?- situacion(a, g, Z)

Z= norte

5) Dado el siguiente resumen del culebrón televisivo "María Patricia":

"Alejandro y María Patricia están casados, pero María Patricia está enamorada de Carlos Alberto. Carlos Alberto no la quiere porque está casado con Manuela pero Augusto quiere a María Patricia. Augusto compite con Luis José que también está enamorado de María Patricia a pesar de estar casado con Bárbara, de quien está enamorado Alejandro".

- Representa el significado básico de este resumen como hechos en Prolog, utilizando sólo 2 predicados distintos.
- Un matrimonio se va a pique si ambos miembros están enamorados de otras personas en vez del uno del otro. ¿Qué personas forman matrimonios que se van a pique? Escribe la consulta necesaria en Prolog y las respuestas que se obtendrían.
- Una persona está celosa cuando una persona a la que ama es amada por una tercera persona, o bien una persona está celosa cuando está casada con una persona de quien una tercera persona está enamorada. Construye la consulta necesaria para determinar las personas que están celosas y los resultados que se obtendrían.

6) Se dispone de una base de hechos sobre relaciones familiares establecidas mediante el predicado *progenitor(A, B)* que se cumple si y sólo si A es el padre o la madre de B.

Se pide definir un predicado *parentesco(+X, +Y, ?N)* tal que, dadas 2 personas, X e Y, determina si son o no parientes y su grado de parentesco N. Consideraremos que dos personas son parientes si una es ascendiente de otra, o bien si tienen un antepasado común o un sucesor común. Si X e Y no son parientes, N devolverá el valor 0. Si X e Y son parientes, N devolverá el número de aristas existentes entre X e Y en el grafo definido por la relación *progenitor*/2.

Hoja 3

Tema 3.2: Representación del conocimiento. Lógica y Prolog

- 7) Representar en Prolog los siguientes hechos según el paradigma del espacio de estados:

Hay un mono en la puerta de una habitación. En el centro de la habitación hay un plátano colgando del techo. El mono está hambriento y quiere conseguir el plátano pero no alcanza porque está muy alto. En la habitación también hay una ventana y debajo de ella hay una caja que le permitiría alcanzar el plátano si se subiera a ella.

El mono puede realizar las siguientes acciones: andar por el suelo, subirse a la caja, empujar la caja (si el mono está en la misma posición que la caja) y coger el plátano (si está subido encima de la caja y la caja está justo debajo del plátano).

¿Podrá el mono coger el plátano? ¿Cuál sería la consulta que tendríamos que realizar?

- 8) Tres misioneros y tres caníbales están a la orilla de un río que quieren cruzar. Para ello disponen de un bote que tiene como capacidad máxima 2 personas. El objetivo consiste en conseguir que todos acaben en la otra orilla del río sin que en ningún momento los misioneros estén en peligro de ser devorados por los caníbales. Se considera que los misioneros están en peligro cuando, en un determinado lugar, el número de caníbales supera al de misioneros. Se pide:

- a) Representar este problema en Prolog según el paradigma del espacio de estados, teniendo en cuenta que no debe permitirse realizar un traslado de personas de una orilla a otra que conduzca a una situación peligrosa.
- b) Construir un programa en Prolog que determine si es posible alcanzar el estado final desde el inicial y produzca la secuencia de operaciones necesarias, evitando la repetición de estados.
- c) Representar gráficamente el espacio de búsqueda correspondiente al programa que se ha desarrollado.
- d) Plantear la ecuación que permitiría calcular el factor de ramificación efectivo correspondiente al árbol de búsqueda del apartado c).

- 9) Un granjero se encuentra en la orilla izquierda de un río junto con un lobo, una cabra y una col. Dispone de un bote en el que sólo puede transportar una única cosa cada vez. El granjero pretende transportar al lobo, la cabra y la col al otro lado del río, utilizando el bote. Sin embargo, debe tener cuidado y no dejar solos en una orilla al lobo y a la cabra porque el lobo se comería a la cabra. Tampoco puede dejar solas a la cabra y la col porque la cabra se comería la col. ¿Cómo conseguirá el granjero trasladar todo a la margen derecha del río?

Se pide representar el problema en Prolog según el paradigma del espacio de estados eligiendo una representación adecuada de estados y operadores. Construir un predicado que determine si es posible alcanzar el estado objetivo a partir del estado inicial y que produzca la secuencia de operadores que se ha seguido para alcanzarlo.

Hoja 3**Tema 3.2: Representación del conocimiento. Lógica y Prolog**

10) Se tienen 3 bloques A, B, y C, y 3 posiciones sobre una mesa P, Q y R. El estado de este micro-mundo de bloques viene dado por una lista de relaciones *sobre*, establecidas entre bloques que están directamente uno encima del otro, o entre bloques y posiciones de la mesa, si el bloque se apoya directamente sobre la mesa. Por ejemplo:

$$\begin{array}{ccc} A & & \\ B & \text{---} & C \\ P & Q & R \end{array} \quad [\text{sobre}(a, b), \text{sobre}(b, p), \text{sobre}(c, r)]$$

Se dispone de un robot que puede realizar 2 acciones: mover un bloque para colocarlo sobre otro bloque y mover un bloque para colocarlo sobre la mesa. Para cualquiera de las dos acciones se necesita que el bloque a mover esté libre (no tenga nada encima). También se necesita que el lugar de destino (bloque o posición en la mesa) esté libre. Cada acción realizada modifica el estado del micro-mundo.

Se pide construir un programa en Prolog que, dados un estado inicial y final, determine si es posible alcanzar el estado final y devuelva la secuencia de acciones necesarias. Para simplificar, se considera que el estado del micro-mundo de bloques está representado mediante una lista ordenada formada por tres relaciones *sobre*, en la cual la relación que describe sobre quién está el bloque A debe preceder a la relación correspondiente sobre el bloque B y ésta a la del bloque C.

11) Sea un micro-mundo formado por 3 habitaciones y un robot aspirador. Hay una habitación a la izquierda, otra en el centro y otra a la derecha, cuyas alfombras pueden estar sucias o limpias. El robot puede estar en cualquiera de las habitaciones y puede ejecutar dos tipos de operaciones: aspirar o moverse. La operación de aspirar requiere que la habitación en la que se encuentra el robot esté sucia y su resultado es que dicha habitación pasa a estar limpia. La operación mover tiene dos opciones: mover hacia la izquierda, que requiere que haya alguna habitación a la izquierda de aquella en la que se encuentra el robot, y mover hacia la derecha que requiere la existencia de alguna habitación situada a la derecha. En cualquier caso, los movimientos serán elementales, es decir, sólo permitirán el paso a la habitación contigua a la actual. Se pide:

- a) Construir un programa en Prolog que, dados un estado inicial y un estado final de este micro-mundo, determine si es posible alcanzar el estado final desde el inicial y devuelva la secuencia de operaciones necesarias, evitando la repetición de estados.
- b) ¿Hay algún estado inicial que garantice que cualquier estado final será alcanzable?
- c) Suponiendo que partimos de un estado inicial en el que están sucias las habitaciones de la izquierda y de la derecha, encontrándose el robot en la habitación de la derecha, y que queremos llegar a un estado final en el que las tres habitaciones están limpias y el robot se encuentra en la habitación del centro, establece cuál es el espacio de estados y representa el espacio de búsqueda correspondiente al programa que has desarrollado.

Hoja 3

Tema 3.2: Representación del conocimiento. Lógica y Prolog

12) Se tienen 2 bloques rectangulares A y B, y 2 posiciones sobre una mesa, P y Q. El estado de este micro-mundo de bloques viene dado por una lista de relaciones *sobre*, establecidas entre bloques que están directamente uno encima del otro, o entre bloques y posiciones de la mesa (si el bloque se apoya directamente sobre la mesa), junto con una indicación de la orientación que tiene cada bloque, vertical u horizontal. Por ejemplo:

$\frac{A}{B}$	
$\frac{P}{Q}$	$[sobre(a, b), sobre(b, p), orientación(a, horizontal), orientación(b, horizontal)]$

Se dispone de un robot que puede realizar 3 acciones: mover un bloque para colocarlo sobre otro bloque, mover un bloque para colocarlo sobre la mesa y girar un bloque, cambiando su orientación. Para cualquiera de las dos acciones de mover se necesita: que el bloque a mover esté libre (no tenga nada encima) y que el lugar de destino (bloque o posición en la mesa) esté libre. Además, si vamos a colocar un bloque sobre otro, necesitamos que ambos tengan la misma orientación, o bien que el bloque de encima esté en vertical y el de abajo en horizontal. Nunca podremos colocar un bloque horizontal sobre uno vertical. Para poder girar un bloque se necesita que no tenga nada encima y que esté sobre otro bloque (no directamente sobre la mesa). Cada acción realizada modifica el estado del micro-mundo.

Se pide construir un programa en Prolog que, dados un estado inicial y un estado final, determine si es posible alcanzar el estado final desde el inicial y devuelva la secuencia de acciones necesarias. Para simplificar, se considera que la lista que representa el estado está compuesta por dos relaciones *sobre* y dos relaciones *orientación*, y está ordenada, figurando antes la información sobre el bloque A que la del bloque B.

13) Considérese una versión miniatura del puzzle de 8, conocida como el puzzle de 3, con los siguientes estados inicial y objetivo:

<u>Inicial</u>	<u>Objetivo</u>
- 1	1 2
3 2	- 3

Se asume que el hueco puede moverse hacia la derecha, hacia la izquierda, hacia arriba y hacia abajo, siempre y cuando se mantenga dentro de los límites del puzzle. Se pide:

- a) Representar este problema en Prolog según el paradigma del espacio de estados.
- b) Construir un programa en Prolog que determine si es posible alcanzar el estado final desde el inicial y devuelva la secuencia de operaciones necesarias, evitando la repetición de estados.
- c) Representar gráficamente el espacio de búsqueda correspondiente al programa que se ha desarrollado.