Ejercicios Representación
Grado en Robótica. Curso 2021-2022.
Departamento de Electrónica y Computación

1. Ejercicio 1

Una fábrica dispone de una cadena automatizada para el pintado de un determinado tipo de objetos. Cada objeto debe pintarse de un único color, de entre varios disponibles, mediante la aplicación de una pistola de pintado que tiene espacio para un único cartucho de pintura. Mientras que el tiempo de pintado es constante (y, de hecho, puede considerarse despreciable), los cartuchos deben instalarse y desinstalarse de la pistola según un procedimiento, también automático, que invierte una cantidad de tiempo conocida y que depende del color del cartucho anterior y del nuevo a instalar y que se especifica, por ejemplo, con una tabla como la que se muestra a continuación:

Color anterior	Color siguiente	Tiempo (min.)
vacío	azul	6
azul	azul	11
azul	rojo	3
azul	verde	9
vacío	rojo	11
rojo	azul	12
rojo	rojo	4
rojo	verde	10
...	...	...

Tabla 1: Tiempos de recambio entre cartuchos de diferente color

donde "vacío" representa el caso en el que previamente no hay instalado ningún cartucho.

Se pide formalizar en lenguaje de marcos los conceptos necesarios para representar el problema.

2. Ejercicio 2

Los domobots son robot domóticos, es decir, robots que realizan tareas domésticas (por ejemplo, son capaces de transportar comida, ropa... desde un punto de una vivienda a otro, de abrir la puerta de la nevera, de la lavadora o del lavavajillas y poner en marcha tales electrodomésticos). El grupo de Agentes Inteligentes de la Universidad de Santiago de Compostela ha desarrollado el robot Brutus MP4/22, el primer domobot especializado en cambiar las bombillas de las lámparas de una vivienda. El robot, cuando se encuentra en una habitación, es capaz de detectar si una bombilla está fundida, acercar una escalera y coger una bombilla nueva, situarse debajo de la lámpara y subirse en la escalera para cambiar la bombilla estropeada.

Además, para la resolución del problema, se asumen las siguientes consideraciones:

- Una habitación es una cuadrícula de 10×10 celdas; el robot, la bombilla de repuesto y la escalera están siempre en alguna de estas celdas (esto es, suponemos que los tres elementos inicialmente están sobre el suelo).

- Asumimos que el robot, mediante su sofisticado sistema de visión, ha detectado una lámpara fundida y conoce la localización de la escalera y de una bombilla nueva. El robot entonces puede realizar varias operaciones: moverse de una celda a otra contigua, coger una bombilla nueva (sólo si ambos se encuentran en la misma celda), coger la escalera (sólo si ambos se encuentran en la misma celda), empujar la escalera de una celda a otra contigua (sólo si ambos se encuentran en la misma celda) y subirse a la escalera (sólo si ambos se encuentran en la misma celda).

- Para poder cambiar la bombilla necesita que la lámpara, el robot y la escalera se encuentren en la misma celda y el robot esté subido a la escalera; sólo moverá la escalera cuando haya cogido la bombilla de repuesto previamente; una vez que haya cambiado la bombilla se bajará de la escalera y se quedará en la misma casilla finalizando su programa de trabajo.

- Puede haber varios domobot y escaleras en la habitación para arreglar varias lámparas.

Se pide formalizar en lenguaje de marcos los conceptos necesarios para representar el problema. Utilizando la representación anterior, ¿cómo se representaría un problema con un estado inicial donde hay dos robots, una única escalera, dos lámparas fundidas y cuatro bombillas?

3. Ejercicio 3

Una empresa de tecnología industrial desea automatizar el proceso de decisión empleado para dirigir el movimiento de un ascensor en grandes edificios. En los problemas que pretenden resolver, se considera la presencia de varias personas distribuídas por un mismo edificio que, en un momento dado, solicitan la atención del ascensor. Por lo tanto, se asume que, de cada persona, se conoce el piso en el que se encuentra, así como el piso al que desea ir. Se pide:

1. Empleando la lógica de predicados, indicar qué predicados sirven para describir cada estado.

2. Pon un ejemplo de un estado inicial y un estado final con tres personas.

4. Ejercicio 4

La matrioska o matrioska son unas muñecas tradicionales rusas creadas en 1890, cuya originalidad consiste en que se encuentran huecas por dentro, de tal manera que en su interior albergan una nueva muñeca, y ésta a su vez a otra, y ésta a su vez otra, en un número variable que puede ir desde cinco hasta el número que se desee, aunque es raro que pasen de veinte. Se caracterizan por ser multicolores, o por la presencia de elementos decorativos en la pintura tales como jarrones o recipientes sostenidos por las muñecas, pero en la práctica pueden identificarse por su tamaño. Las matrioskas pueden verse como un juego. Las matrioskas pueden estar abiertas (el cuerpo separado de la cabeza), o cerradas. Una matrioska puede contener a otra matrioska. Los estados inicial y final pueden ser cualquier configuración en la que todas las matrioskas están: (i) o bien cerradas en la mesa y/o unas dentro de otras, (ii) o bien abiertas encima de la mesa. Por tanto, unas matrioskas que inicialmente tienen la configuración que se muestra en la Figura 1 no tendrían una configuración correcta, puesto que hay unas matrioskas abiertas dentro de otras.

Figura 1: Configuración incorrecta de matrioskas

Una matrioska sólo se puede abrir si está encima de la mesa. Una matrioska cerrada que está encima de la mesa puede meterse en otra si la segunda está abierta, encima de la mesa y es más grande que la primera. Se pide:

1. Formalizar el enunciado con lógica de predicados. Se valorará que la formalización dada sea fácilmente escalable a cualquier número de matrioskas.

2. Formalizar un problema de 5 matrioskas. En el estado inicial, la más grande está cerrada y vacía encima de la mesa, y las demás están todas cerradas dentro de la de tamaño inmediatamente superiorde la que le corresponde (excepto para la cuarta más grande que está encima de la mesa,puesto que la másgrande está vacía). En su configuración final están todas cerradas y dentro de la de tamaño inmediatamente superior.

5. Ejercicio 5

En el fondo de un viejo armario descubres una nota escrita por un pirata famoso por su sentido del humor y su afición por los acertijos lógicos. En la nota dice que ha escondido un tesoro en algún lugar de una propiedad. El pirata enumera cinco enunciados todos ellos verdaderos y te reta a que descubras dónde está el tesoro. He aquí los enunciados:

- Si la casa está cerca de un lago, el tesoro no está en la cocina.
- Si el árbol de la entrada es un olmo, el tesoro está en la cocina.
- La casa está cerca de un lago.
- El árbol de la entrada es un olmo o el tesoro está enterrado debajo del mástil.
- Si el árbol de la entrada es un roble, el tesoro está en el garaje.

Utilizando lógica proposional, deducir dónde está el tesoro.

6. Ejercicio 6

Una empresa desea diseñar un sistema de razonamiento automático para tomar decisiones sobre la canalización de las aguas a través de su red de distribución. En concreto, su red de distribución está constituida por una cantidad arbitraria de depósitos (denominados genéricamente con las letras A,B,. . .) de los que salen canales que discurren en paralelo hasta un número igual de destinos —identificados con los números 1, 2. . . Entre los canales, puede haber tuberías que desvían todo el agua que llega de un canal hasta otro inmediatamente adyacente, en cualquier sentido.

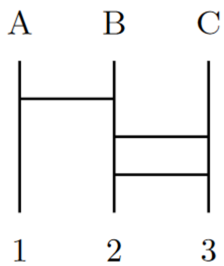

Figura 2: Modelo de distribución de aguas entre tres depósitos y destinos

Por ejemplo, la figura 2 muestra el caso de tres canales entre tres depósitos (A, B y C), hasta tres destinos diferentes (1, 2 y 3), entre los que se han dispuesto hasta tres tuberías entre canales adyacentes. **Se debe asumir que las tuberías siempre están abiertas y que, por ello, conducirán todo el caudal de agua que les llegue**

desde un canal hasta otro. Asimismo, sólo está abierto un único depósito al mismo tiempo. Por lo tanto, si se abriera el depósito de A, todo el agua se conduciría hasta el destino marcado con el número 2, pasando primero por la tubería que va de A a B, desde donde llegaría a C por la segunda tubería y volvería, nuevamente, a B por la última tubería; o si nada más que se abrieran las llaves de paso del depósito C, entonces el agua llegaría únicamente hasta el destino 3; por último, el agua desde B llegaría a 1.

Se pide:

1. Modelizar la red de canales y tuberías de la figura 2 con el uso de la lógica proposicional, detallando los literales y reglas necesarias para ello.

2. ¿Es posible deducir automáticamente hasta qué destino llegaría el agua desde el depósito B con la modelización del apartado anterior? Si es así, ¿cómo?, y si no es así, ¿por qué?

3. ¿Es razonable el uso de la lógica proposicional para la modelización de redes de distribución arbitrariamente grandes? Si o no, y por qué

4. Modelizar la red de canales y tuberías de la figura 2 con el uso de la lógica de predicados, detallando los predicados y reglas necesarias para ello

7. Solución ejercicio 1

Los elementos fundamentales para la representación de cada estado, son únicamente: la `pistola`, indistinguible del cartucho mismo puesto que siempre se asumirá que está cargada con alguno; cada `objeto` a pintar, naturalmente, sin los que no habría problema y, por último, la tabla de `coste` para cambiar un cartucho por otro. Así, por lo tanto, la representación con marcos que se sugiere es la que se muestra a continuación.

Nótese, en primer lugar, que en el marco de la tabla 2, el `Vacío` es considerado como cualquier otro color. No hay de hecho ningún motivo para no hacerlo así. Además, la `pistola` mantiene el número de pintados que aún quedan disponibles en su cartucho que, por defecto, serán 0 puesto que ése es el número de pintados que se hacen en `Vacío`. Por último, se entiende que cada `pistola` tiene una capacidad máxima de pintados.

Pistola	
`es-un:`	
Atributo	**Posibles valores/Valor**
color	SÍMBOLO/“Vacío”
número	ENTERO/0
capacidad	CAPACIDAD/0

Tabla 2: Representación del concepto `Pistola`.

Para la definición de cada `objeto` se emplea el marco de la tabla 3:

Objeto	
`es-un:`	
Atributo	**Posibles valores/Valor**
color	SÍMBOLO
número	ENTERO

Tabla 3: Representación del concepto `Objeto`.

donde se ha tenido en cuenta una simplificación notable que reduce significativamente el espacio de estados: *en vez de representar cada objeto separadamente, éstos se agrupan por su color*, puesto que en la solución óptima no tiene sentido dejar cartuchos a medias con más objetos de pintar del mismo color. Por lo tanto, cada estado mantiene sobre sus `objetos`, el `número` de ellos que deben pintarse de cada `color`.

El último marco representado en la tabla 4 es trivial, y debe servir únicamente para registrar el coste de cambiar de un cartucho de un color a otro, de modo que la siguiente definición es auto-explicativa y donde t registra el tiempo que se tardará en reemplazar un cartucho de color_1 por otro de color_2:

Coste	
`es-un:`	
Atributo	**Posibles valores/Valor**
color_1	SÍMBOLO
color_2	SÍMBOLO
t	ENTERO

Tabla 4: Representación del concepto `Coste`.

8. Solución ejercicio 2

Se piden los marcos necesarios para resolver el problema. Para ello es suficiente definir 4 marcos, `ROBOT`, `BOMBILLA`, `LAMPARA` y `ESCALERA`. Una de las ventajas de utilizar marcos es el poder hacer una jerarquía de clases. En este caso, los cuatro marcos tienen unos atributos comunes por los que es mejor definir una clase abstracta `OBJETO` agrupando todas las características comunes. Cada objeto debe estar situado en la habitación, por tanto, se le pueden definir dos atributos, posición x e y con lo que quedan perfectamente definida su posición y se evita definir un marco para las casillas. Se supone que el origen de coordenadas $(1, 1)$ está en la esquina superior izquierda. Por tanto, los marcos pedidos para representar el sistema son los siguientes:

OBJETO			
`es-un:`			
Atributo	**Posibles valores/Valor**	**Valor omisión**	**Descripción**
x	entero {1,...,10}		posición x
y	entero {1,...,10}		posición y
id	símbolo	0	identificador único del objeto
estado	símbolo	libre	estado en el que está cada objeto

Tabla 5: Representación del concepto `OBJETO`.

En el marco `ROBOT` se especializa el atributo `estado` para representar los posibles estados en que puede estar el domorobot. Estos son, `libre`, estado inicial; `bombilla`, cuando ha detectado una lámpara fundida y se dirige a recoger una bombilla de recambio; `escalera`, cuando coge la bombilla de recambio y se dirige a coger una escalera libre; `lampara`, una vez que tiene la escalera debe dirigirse a la lámpara fundida para cambiarla. De esta forma se controla que los domorobots sigan el orden de acciones pedido. Además, tendrá dos atributos más para indicar las coordenadas x e y del objeto al que debe dirigirse.

ROBOT			
`es-un: OBJETO`			
Atributo	**Posibles valores/Valor**	**Valor omisión**	**Descripción**
$objx$	entero {1,...,10}		posición x a la que debe dirigirse
$objy$	entero {1,...,10}		posición y a la que debe dirigirse
id	símbolo	0	identificador único del objeto
estado	símbolo {libre bombilla escalera lámpara}	libre	controla el orden de las acciones

Tabla 6: Representación del concepto `ROBOT`.

En los siguientes marcos el atributo `estado` tomará el valor del identificador del robot que lo coge o se le asigna para arreglar para evitar que interfieran unos domorobots con otros. En el caso del marco `LAMPARA` se utilizará el valor `fundida` para indicar que hay que arreglarla. Al ser símbolos generales no es necesario especializar la definición de la clase padre.

LAMPARA
`es-un: OBJETO`

Tabla 7: Representación del concepto `LAMPARA`.

ESCALERA
`es-un: OBJETO`

Tabla 8: Representación del concepto `ESCALERA`.

BOMBILLA
`es-un: OBJETO`

Tabla 9: Representación del concepto `BOMBILLA`.

Para representar el estado inicial pedido, habría que añadir dos instancias de `ROBOT`, con identificadores diferentes y sus posiciones x e y iniciales; una instacia de `ESCALERA`, con su identificador y posición inicial; dos instancias de `LAMPARA`, con el atributo estado fundida, su identificador y su posición *x* e *y*; y cuatro instancias de `BOMBILLA`, cada una con un identificador y una posición *x* e *y*.

9. Solución ejercicio 3

1. Existen varias formas de intentar resolver este problema, pero la más eficiente consiste en determinar qué personas ya han llegado a sus destinos. Estos hechos podrían almacenarse en los predicados `served(persona)` y `not-served(persona)`. Obviamente, es preciso conocer la posición actual de cada persona (predicado `origin(persona, planta)`) y a qué planta desea llegar (predicado `destin(persona, planta)`), así como distinguir si se encuentra dentro de un ascensor (predicado `boarded(persona)`) o no —predicado `not-boarded(persona)`.Por último, para poder determinar el movimiento del ascensor, bastará con conocer su posición (`lift-at(planta)`), así como la relación de precedencia entre las plantas, para lo que basta el predicado `above(planta-1, planta-2)`.

2. En la definición de un estado inicial, imagínese que Mónica, Gema y Fernando desean reunirse todos en la planta número 3. Mientras que Mónica y Gema están en la planta 1, en el estado inicial dispondremos a Fernando en la planta 5 (de modo que nadie está dentro del ascensor):

```
origin(Mónica, 1)  origin (Gema, 1)  origin(Fernando, 5)
destin(Mónica, 3)  destin(Gema, 3)  destin(Fernando, 3)
not-boarded(Mónica)  not-boarded(Gema)  not-boarded(Fernando)
```

   y el ascensor está, por ejemplo, en la planta 4:

```
lift-at(4)
```

   y la relación de precedencia entre las plantas, en un edificio con 7 plantas por ejemplo, se establece con:

```
above(1,0)  above(2,1)  above(3,2)
above (4,3)  above(5,4)  above(6,5)
above (7,6)
```

   como ninguno de ellos está en su planta destino, debe añadirse:

```
not-served(Mónica)  not-served(Gema)  not-served(Fernando)
```

   Por último, el estado final consiste, simplemente, en solicitar que todas las personas lleguen a sus destinos:

```
served(Mónica)  served(Gema)  served(Fernando)
```

10. Solución ejercicio 4

1. Para describir el dominio, es necesario describir los tipos de los objetos y los predicados del dominio. En este caso, tenemos un único objeto MATRIOSKA que hace referencia a las figuritas rusas. En cuanto a los predicados, tenemos los siguientes predicados:

   - (`closed` $\langle m \rangle$): indica que la matrioska $\langle m \rangle$ está cerrada
   - (`on-table`$\langle m \rangle$): indica que la matrioska $\langle m \rangle$ está encima de la mesa

- (`bigger` $\langle m2\rangle$ $\langle m1\rangle$): indica que la matrioska $\langle m2\rangle$ es mayor que la matrioska $\langle m1\rangle$
- (`empty` $\langle m\rangle$): indica que la matrioska $\langle m\rangle$ no contiene ninguna otra matrioska
- (`in` $\langle m1\rangle$ $\langle m2\rangle$): indica que la matrioska $\langle m1\rangle$ está dentro de la matrioska $\langle m2\rangle$

2. Utilizando el listado de predicados anterior, el estado inicial quedaría como:

```
(on-table matri4)  (on-table matri5)
(in matri1 matri2)  (in matri2 matri3)  (in matri3 matri4)
(empty matri1)  (empty matri4)  (empty matri5)
(closed matri1)  (closed matri2)  (closed matri3)  (closed matri4)  (closed matri5)
(bigger matri2 matri1)   (bigger matri3 matri1)   (bigger matri4 matri1)
(bigger matri5 matri1)   (bigger matri3 matri2)   (bigger matri4 matri2)
(bigger matri5 matri2)   (bigger matri4 matri3)   (bigger matri5 matri3)
(bigger matri5 matri4)
```

y el estado final:

```
(in matri1 matri2)
(in matri2 matri3)
(in matri3 matri4)
(in matri4 matri5)
(closed matri5)
```

Cabe destacar que en esta meta sólo se especifica como meta que esté cerrada la matrioska `matri5`. Dado que el resto de matrioskas están unas dentro de otras, por las características del dominio se cumplirá que todas están cerradas.

11. Solución ejercicio 5

Para resolver el ejercicio, en primer lugar es necesario identificar los literales o variables proposionales que se emplearán para posteriormente deducir dónde está el tesoro. En este caso particular, las proposiones que se utilizan son:

- p: La casa está cerca de un lago
- q: El tesoro está en la cocina
- r: El árbol de la entrada es un olmo
- s: El tesoro está enterrado debajo del mástil
- t: El árbol de la entrada es un roble
- v: El tesoro está en el garaje

donde cada uno de estas variables proposicionales puede tomar el valor de verdadero o falso en la deducción posterior. A continuación, podemos formalizar cada uno de los enunciados que aparecen como:

- $p \rightarrow \neg q$: Si la casa está cerca de un lago, el tesoro no está en la cocina. Se puede comprobar por **modus ponens**, que si la variable proposicional p es verdadera, es decir, la casa está cerca de un lago, entonces el tesoro no está en la cocina tiene que ser verdadero, es decir, $\neg q$ tiene que ser verdadero. Para ello q, es decir, la variable que representa que el tesoro está en la cocina debe ser falso.

- $r \rightarrow q$: Si el árbol de la entrada es un olmo, el tesoro está en la cocina. La explicación de esta regla es análoga a la anterior, teniendo en cuenta las nuevas variables involucradas.
- p: La casa está cerca de un lago. En este caso, tenemos un enunciado que nos dice que una de nuestras variables es directamente verdadera en nuestra base de hechos iniciales.
- $r \vee s$: El árbol de la entrada es un olmo o el tesoro está enterrado debajo del mástil.
- $t \rightarrow v$: Si el árbol de la entrada es un roble, el tesoro está en el garaje.

El enunciado nos dice que todas estas afirmaciones son verdaderas, luego la expresión:

$R1 : (p \rightarrow \neg q) \wedge (r \rightarrow q) \wedge p \wedge (r \vee s) \wedge (t \rightarrow v)$

también tiene que serlo. Como es un *and* lógico, eso significa que cada una de la expresiones que lo componen tienen que ser verdadero. Vamos a tratar de simplificar la expresión $R1$, sustituyendo las variables por sus valores que ya conocemos. Si en la expresión $R1$ la variable proposicional p es verdadera, entonces, la expresión queda reducida a:

$R2 : \neg q \wedge (r \rightarrow q) \wedge (r \vee s) \wedge (t \rightarrow v)$

donde hemos eliminado la cláusula p que es verdadera y la regla $(p \rightarrow \neg q)$ queda simplemente como $\neg q$. Como se puede ver, en esta nueva expresión $R2$, $\neg q$ tiene que ser verdadera, para que $R2$ también lo sea, lo que implica que q es falsa. Por lo tanto, hasta ahora sabemos que p, es decir, la casa está cerca del lago, y que $\neg q$, es decir, el tesoro no está en la cocina. Hemos descartado el primer lugar donde no está el tesoro: en la cocina. Haciendo estas sustituciones, $R2$ queda como:

$R3 : \neg r \wedge (r \vee s) \wedge (t \rightarrow v)$

Nótese que para llegar a esta nueva expresión $R3$, se ha tenido en cuenta que $(r \rightarrow q) \equiv (\neg r \vee q)$. Este hecho puede comprobarse fácilmente haciendo algunas sustituciones:

- Con la regla $r \rightarrow q$ tenemos que si r es verdadero, q necesariamente también debe serlo. Eso lo podemos comprobar también con la expresión, $\neg r \vee q$, donde si r es verdadero, la única posibilidad para que la expresión $\neg r \vee q$ sea verdadera, es que q también lo sea.
- Sin embargo, la regla $r \rightarrow q$ no nos dice nada sobre q en el caso de que r sea falsa. En ese caso, q podría ser tanto verdadera como falsa. Esto también se comprueba con la expresión $\neg r \vee q$, donde si r es falsa, q podría ser tanto verdadera como falsa.

De la deducción anterior sabemos que q es falsa, por lo tanto, la expresión $(r \rightarrow q) \equiv (\neg r \vee q)$ queda como $\neg r$ en $R3$. Para que $\neg r$ sea verdadera, r tiene que ser falsa. Luego tenemos que el árbol de la entrada no es un olmo, que se une a los hechos que ya teníamos como ciertos: la casa está cerca de un lago, y el tesoro no está en la cocina. Si r es falsa, la expresión $R3$ se reduce a:

$R4 : s \wedge (t \rightarrow v)$

En esta nueva expresión, se ha sustituido $(r \vee s)$ por simplemente s. Nótese que si r es falsa, que la expresión $(r \vee s)$ sea verdadera recae únicamente en el valor final que tome s, de ahí que $(r \vee s)$ pase a ser simplemente s. En $R4$, como antes, cada uno de los elementos que componen el *and* lógico tienen que ser verdaderos para que toda la expresión sea verdadera. Eso significa que s tiene que ser verdadera, lo que indica que el tesoro está enterrado debajo del mástil. Nótese que para llevar a cabo esta deducción no ha sido necesario utilizar la última regla del enunciado.

12. Solución ejercicio 6

1. La modelización de la red de distribución se hace modelando, por separado, sus tres componentes: los depósitos, tuberías y destinos. Por otra parte, los canales no tienen por qué modelizarse realmente y basta con describir secuencialmente los elementos que contienen. Puesto que el agua siempre circula en la misma dirección, puede asumirse que los canales están divididos en pasos o etapas como muestra la figura 3, numerados ascendentemente desde los depósitos hasta los destinos.

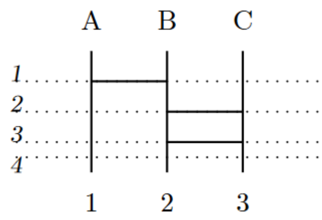

Figura 3: Los canales se modelizan en pasos o etapas

En concreto, los canales se han dividido en cuatro pasos diferentes. Los tres primeros sirven para identificar la posición en la que se han dispuesto tuberías entre canales. El último tiene un propósito diferente y sirve para determinar el punto de cada canal en el que no hay más tuberías y, por lo tanto, están directamente conectados con los destinos o salidas. Su propósito se entenderá mejor al presentar las reglas que rigen el movimiento del agua en la red de la figura 2.

Por lo tanto, los literales o variables proposicionales necesarios para modelizar la red son:

- **Depósitos** Los depósitos se identificarán con el literal d_X donde X representa el nombre de un depósito. En concreto, en la figura 2 hay hasta tres depósitos: d_A, d_B y d_C.
- **Destinos** Análogamente, cada destino se modelará con literales s_Y. En el caso de la figura 2 habrá hasta tres: s_1, s_2 y s_3.
- **Tuberías** Las tuberías sirven para conducir el agua y, por lo tanto, su comportamiento se modelará con el uso de reglas. Cada tubería desde un canal i hasta otro canal adyacente, j, en el paso k, se transcribe como: $p_{ik} \rightarrow p_{j(k+1)}$. El hecho de que el consecuente de cada regla incremente el valor del paso sirve, precisamente, para indicar que el agua progresa en el siguiente canal. Además, puesto queel agua puede transcurrir en cada tubería en cualquier sentido (según venga por el primer o segundo canal), la misma tubería generará, además, otra regla del modo $p_{jk} \rightarrow p_{i(k+1)}$. En concreto, las tres tuberías de la figura 2 se representan en lógica proposicional con las seis reglas siguientes:

$$
\begin{aligned}
&p_{A1} \rightarrow p_{B2}\\
&p_{B1} \rightarrow p_{A2}\\
&p_{B2} \rightarrow p_{C3}\\
&p_{C2} \rightarrow p_{B3}\\
&p_{B3} \rightarrow p_{C4}\\
&p_{C3} \rightarrow p_{B4}
\end{aligned}
$$

Sin embargo, esta modelización no es suficiente ni para mostrar cómo llega el agua hasta la red de distribución,ni cómo sale de ella. En realidad, hacen falta reglas adicionales para iniciar el movimiento del agua cuando se abre una llave en uno de los depósitos (¡y sólo uno!), y para indicar cómo el caudal se acumula en uno de los destinos o salidas:

- **Entrada** La entrada del agua se hace, simplemente, con reglas que indican que cuando se usa un determinado depósito (identificado con un literal de la forma d_X), el agua llega necesariamente al primer paso en el que haya una tubería de ese mismo canal [1]

Así, las reglas que rigen la entrada de agua en la figura 2 son:

$d_A \rightarrow p_{A1}$
$d_B \rightarrow p_{B1}$
$d_C \rightarrow p_{C2}$

- **Salida** La salida se rige, igualmente, por reglas que determinan el punto por el que saldrá el agua. Sin embargo, aquí hay un pequeño detalle que debe resolverse: si nada más se numeran los puntos en los que hay tuberías, cuando el agua llega a la última tubería no sería posible saber si es que cambiará de canal o, por el contrario, progresará por el mismo canal hacia su salida. Es precisamente por este motivo por el que, como se anticipaba antes, es preciso disponer de un paso adicional (el cuarto en el caso de la figura 3) para distinguir un caso del otro: en el paso tres, el agua cambia de canal y en el cuatro se dirige hacia la salida.

  Las reglas para la descripción de la salida del agua son análogas a las de la entrada y resultan ser [2]:

  $p_{A2} \rightarrow s_1$
  $p_{B4} \rightarrow s_2$
  $p_{C4} \rightarrow s_3$

2. Por supuesto que se puede. Para ello, basta con usar repetidamente el **modus ponens**. En concreto, si se abre la llave de paso en el depósito B, entonces d_B es cierto.

   Usando la regla del **modus ponens** sobre la regla $d_B \rightarrow p_{B1}$, que rige la entrada de agua en todo el sistema, se concluye necesariamente que el agua llega hasta el primer paso del canal B —que es precisamente lo que indica el consecuente de la regla usada: p_{B1}. A continuación, la regla $p_{B1} \rightarrow p_{A2}$ que modeliza la primera tubería con la que se encuentra el agua en el canal B, sirve para deducir, automáticamente (nuevamente en virtud del **modus ponens**), p_{A2} o, lo que es lo mismo, que el agua transcurre ahora por el segundo paso del canal A. Por último, la regla de salida $p_{A2} \rightarrow s_1$, servirá para concluir de la misma manera s_1 o, equivalentemente, que todo el agua llega hasta el primer destino.

3. No lo es en absoluto. Mientras que la ventaja de la lógica proposicionales que es decidible, lo cierto es que adolece de capacidad de generalización. En concreto, todas las afirmaciones sobre grandes colecciones de individuos resultan en la definición de muchos literales y reglas esencialmente repetitivos.

   Este problema se resolverá, precisamente, con el uso de la lógica de predicados en el siguiente apartado.

4. Naturalmente, para la modelización con lógica de predicados, deben tenerse en cuenta los mismos elementos y comportamientos identificados en el apartado 1. Sin embargo, en vez de usar literales, ahora es preciso distinguir predicados (que describirán el qué) y sus relaciones o el cómo, con el uso de reglas.

   En realidad, ahora el esfuerzo de modelización es un poco mayor porque es preciso hacer una descripción general de cualquier red de distribución. En concreto:

   - **Depósitos** Los depósitos se identificarán con el predicado $Deposito(x)$ donde x representa el nombre de un depósito. En concreto, en la figura 2 hay hasta tres depósitos: $Deposito(A)$, $Deposito(B)$ y $Deposito(C)$.
   - **Destinos** Análogamente, cada destino se modelará conpredicadosde la forma $Destino(y)$. En el caso de la figura 2 habrá hasta tres: $Destino(1)$, $Destino(2)$ y $Destino(3)$.

[1] En particular, nótese que la entrada de agua desde el depósito C se puede hacer directamente hasta el segundo paso, puesto que antes no hay ninguna tubería que desvíe el movimiento del agua

[2] Como ocurriera con las entradas, nótese ahora que la salida a 1 se puede modelizar en el canal A desde el paso 2 y es que a partir de él ya no hay tuberías que desvíen el curso de su caudal

- **Tuberías** Como ocurriera en el caso de la lógica proposicional, las tuberías se describirán con reglas que determinarán el flujo del agua. En concreto, una tubería que progresa desde un canal i hasta otro canal j, en el paso k, se modelizará como: $[Agua(i,k) \wedge Tuberia(i,j,k) \wedge Succ(k,k') \rightarrow Agua(j,k')]$, donde $Agua(i,k)$ representa la observación de que el agua ha llegado hasta el paso k del canal i, y $Tuberia(i,j,k)$ representa que hay una tubería que vincula los canales i y j precisamente en el paso k. Asimismo, es preciso observar que la lógica de predicados no tiene una representación aritmética implícita. Por lo tanto, es preciso modelar la relación de sucesión entre los números naturales (que representan las fases o etapas por las que progresa el agua) con un predicado dedicado, $Succ(k,k')$, que representa el hecho de que k' es el sucesor de k.

  Ahora bien, en lógica de predicados es posible utilizar variables para generalizar la representación del conocimiento. Por lo tanto, en vez de tener tantas reglas como tuberías, basta con una [3]:

  $R1 : \forall i,j,k,k'[Agua(i,k) \wedge Tuberia(i,j,k) \wedge Succ(k,k') \rightarrow Agua(j,k')]$

  que, sin embargo, no es suficiente para representar los dos sentidos de circulación del agua. Por ello, es preciso añadir una segunda regla:

  $R2 : \forall i,j,k,k'[Agua(j,k) \wedge Tuberia(i,j,k) \wedge Succ(k,k') \rightarrow Agua(i,k')]$

- **Canales** A diferencia del caso de la lógica proposicional, aquí es necesario describir explícitamente también los puntos en los que no hay tuberías para modificar el flujo de circulación del agua, esto es, en aquellos casos en los que agua continua transcurriendo por el mismo canal. El motivo es que mientras en el primer apartado era posible especializar cada regla para cada caso (como, por ejemplo, el hecho de que el agua desde C llega hasta el segundo paso inmediatamente), porque de hecho había que describir una regla por tubería, ahora debe hacerse una descripción generalizada

  Por supuesto, podría simplemente usarse la expresión $\neg Tuberia(i,j,k)$, para indicar que el agua no cambia desde el canal i hasta el canal j, pero en este caso, no sería posible utilizar, por ejemplo, deducción hacia detrás puesto que es imperativo que todos los literales estén afirmados. Por lo tanto, se usará un predicado $Progreso(i,k)$ que indica simplemente que el agua continuará en el canal i, cuando se encuentre en el paso k.

  Las reglas que describen el paso del agua es, en este caso:

  $R3 : \forall i,k,k'[Agua(i,k) \wedge Progreso(i,k) \wedge Succ(k,k') \rightarrow Agua(i,k')]$

  Por último, es preciso también crear reglas especiales para describir la entrada y salida del agua en la red de distribución:

  - **Entrada** Si el predicado $Deposito(x)$ sirve para identificar que se abre la llave de paso del depósito especificado, entonces la siguiente regla describe la entrada de agua desde cualquier punto:

    $R4 : \forall x[Deposito(x) \rightarrow Agua(x,1)]$

  - **Salida** Análogamente, si el predicado $Destino(y)$ representa, también, el hecho de que se recibe agua en y,entonces todas las salidas se describen simplemente como:

    $R5 : \forall x[Agua(x,4) \rightarrow Destino(x)]$

    donde el número 4 sirve simplemente para identificar el último paso de la figura 2.

[3] En lo sucesivo, se asume que nuestro sistema computacional está debidamente poblado de tantos predicados $Succ(1,2)$, $Succ(2,3)$,... como haga falta