INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

Ingeniería Técnica en Informática de Sistemas (UNED)

Código de carrera: 53, Código de asignatura: 2097

SEPTIEMBRE 2008 DURACIÓN: 2 hrs. MATERIAL PERMITIDO: Ninguno

Importante: Ponga el nombre en todas las hojas. No sólo se valorará que el resultado sea correcto, sino también la claridad en la exposición de los pasos seguidos en la resolución, que el examen esté compensado y que no incluya errores conceptuales importantes.

Ejercicio 1. (Valoración: 10 / 3)

Realice un estudio comparativo de los siguientes métodos de representación de conocimiento: *Redes Bayesianas* y *Método Clásico de Diagnóstico Probabilista,* éste último también conocido como *Método Probabilista Clásico*. Haga especial énfasis en los siguientes aspectos:

a) Tipo de representación que utilizan
b) Tipo de inferencias que realizan
c) Dominios del mundo real en que se aplican

SOLUCIÓN

a) Los dos métodos utilizan un **grafo** como forma de representación. Mientras que en el *Método Probabilista Clásico* (MPC) el grafo sólo consta de un nodo padre desde el que se traza un arco hacia cada uno de sus nodos hijo, una *Red Bayesiana* (RB) está formada en general por un grafo dirigido acíclico. Aunque el grafo de una red RB no puede tener ciclos, sí puede tener bucles. Dado un camino cerrado en un grafo dirigido, si podemos recorrer el camino siguiendo la dirección de los arcos, tenemos un ciclo; en caso contrario, tenemos un bucle.

Tanto en el MPC como en las RBs, cada nodo del grafo hace referencia a una determinada **variable**. En el MPC, mientras que el nodo padre hace referencia a posibles diagnósticos o enfermedades, cada nodo hijo lo hace a un hallazgo. En RBs, en principio no hay restricciones del estilo de las empleadas en el MPC.

Tanto en el MPC como en las RBs, los **valores** que puede tomar cada variable deben ser exclusivos y exhaustivos. Exclusivos significa que dos o más valores no pueden ser ciertos a la vez. Exhaustivos significa que no puede existir un valor diferente.

En los dos métodos de representación, los **enlaces** expresan relaciones de causalidad o simplemente de dependencia probabilística.

Cada uno de los dos métodos define una serie de **relaciones de independencia** de forma implícita. En el MPC, los hallazgos son condicionalmente independientes entre sí dado el diagnóstico. En las RBs, existen una serie de relaciones de independencia determinadas por la propiedad de separación direccional, que las RBs cumplen por definición: "Dado un nodo X, el conjunto de sus padres, $pa(X)$, separa condicionalmente este nodo de todo otro subconjunto Y en que no haya descendientes de X. Es decir, $P(x \mid pa(x), y) = P(x \mid pa(x))$."

Los dos métodos manejan una serie de **parámetros probabilísticos** asociados a cada nodo: para cada nodo sin padres, la probabilidad a priori de cada uno de sus valores, mientras que para cada nodo con padres, la probabilidad condicional de cada uno de sus valores dada cualquier posible configuración de valores de sus nodos padre.

Finalmente, reseñar que el MPC se puede considerar como un caso particular de RB. De hecho, las RBs aparecieron varios años después que el MPC.

b) En cuanto a la inferencia, en el MPC está basada en el Teorema de Bayes, mientras que en RBs se utilizan esquemas más elaborados, por ejemplo, el paso de mensajes probabilísticos en redes con estructura de poliárbol.

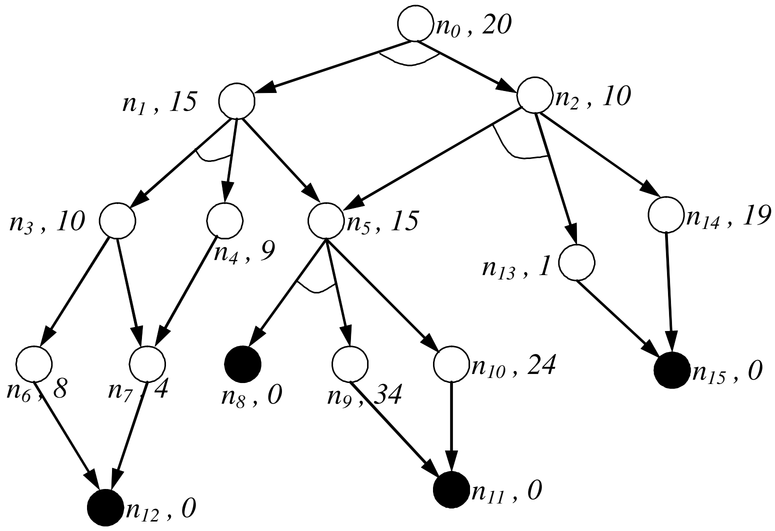
Mientras que en el MPC la inferencia persigue calcular la probabilidad a posteriori del nodo padre dado un conjunto de hallazgos sobre sus nodos hijo, en RBs se pretende calcular la probabilidad a posteriori de cada variable no observada dadas las variables observadas.

c) Los dos métodos se aplican a dominios caracterizados por la presencia de incertidumbre, por ejemplo, el diagnóstico médico. Mientras que el MPC está más enfocado al diagnóstico, las RBs también se pueden aplicar en tareas de predicción.

Ejercicio 2. (Valoración: 10 / 3)

Considere el grafo Y/O (también conocido como A/O) de la figura. Describir paso a paso el desarrollo de la exploración de dicho grafo mediante el algoritmo YO* (también conocido como AO*). Para ello supóngase que el coste de cada arco es 1 y que se tienen los siguientes valores para la función heurística "*h*" de estimación del coste del subgrafo solución óptimo desde cada nodo:

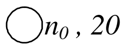
$h(n_0) = 20$	$h(n_3) = 10$	$h(n_6) = 8$	$h(n_9) = 34$	$h(n_{12}) = 0$	$h(n_{15}) = 0$
$h(n_1) = 15$	$h(n_4) = 9$	$h(n_7) = 4$	$h(n_{10}) = 24$	$h(n_{13}) = 1$	
$h(n_2) = 10$	$h(n_5) = 15$	$h(n_8) = 0$	$h(n_{11}) = 0$	$h(n_{14}) = 19$	



Recuerde que n_0 es el nodo inicial y los nodos terminales o meta son n_8, n_{11}, n_{12} y n_{15}. Como criterio para la expansión, en caso de conflicto, se recomienda expandir el nodo de menor índice.

SOLUCIÓN

- **Inicialmente:**



Seguidamente se describe cada ciclo del algoritmo, en cada uno de los cuales primeramente se expande un nodo hoja cualquiera del subgrafo solución parcial que cuelga de n_0 y a continuación, si es necesario, se actualizan hacia arriba los costes de los subgrafos solución parciales que cuelgan de los nodos antepasados del nodo expandido. Para ello utilizaremos un conjunto denominado *S* tal que si sacamos un nodo de *S* y hay que actualizarlo, entonces introducimos en *S* los padres del nodo actualizado. De *S* se irán sacando aquellos nodos que no tengan descendientes en *S*. Inicialmente, el nodo hoja expandido es el único elemento contenido en *S*.

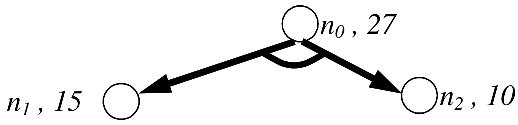
- **Ciclo 1:**

> Se expande n_0.

> La evolución del conjunto *S* es la siguiente:

1) $S = \{n_0\}$
2) Se saca n_0 de *S*. Su nuevo coste es 15+1+10+1 = 27.
3) $S = \{\}$

> La situación final es la siguiente:



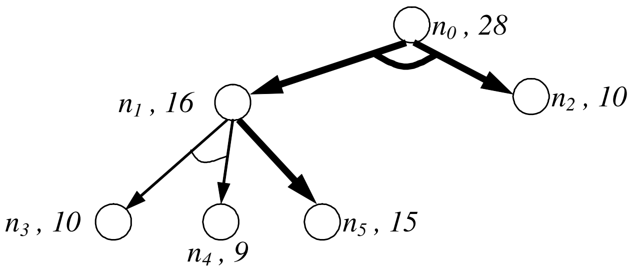
- **Ciclo 2:**

> Se expande n_1, aunque también se podría haber expandido n_2.

> La evolución del conjunto S es la siguiente:

1) $S = \{n_1\}$
2) Se saca n_1 de S. Su nuevo coste es 16. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_1 en S.
3) $S = \{n_0\}$
4) Se saca n_0 de S. Su nuevo coste es 28.
5) $S = \{\}$

> La situación final es la siguiente:



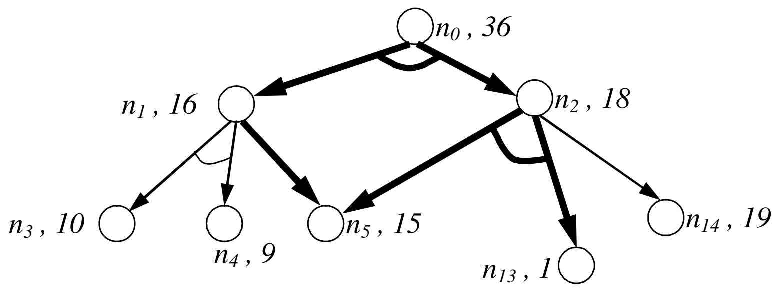
- **Ciclo 3:**

> Se expande n_2, aunque también se podría haber expandido n_5.

> La evolución del conjunto S es la siguiente:

1) $S = \{n_2\}$
2) Se saca n_2 de S. Su nuevo coste es 18. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_2 en S.
3) $S = \{n_0\}$
4) Se saca n_0 de S. Su nuevo coste es 36.
5) $S = \{\}$

> La situación final es la siguiente:



- **Ciclo 4:**

> Se expande n_5, aunque también se podría haber expandido n_{13}.

> La evolución del conjunto S es la siguiente:

1) $S = \{n_5\}$

2) Se saca n_5 de S. Su nuevo coste es 25. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_{15} en S.

3) $S = \{n_1, n_2\}$

4) Se saca n_1 de S, aunque también se podría haber sacado n_2, ya que ni n_2 es descendiente de n_1 ni viceversa. El nuevo coste de n_1 es 21, ya que ha de redirigirse el enlace que apunta al mejor subgrafo que cuelga de n_1. Por tanto, introducimos su padre, n_0, en S.

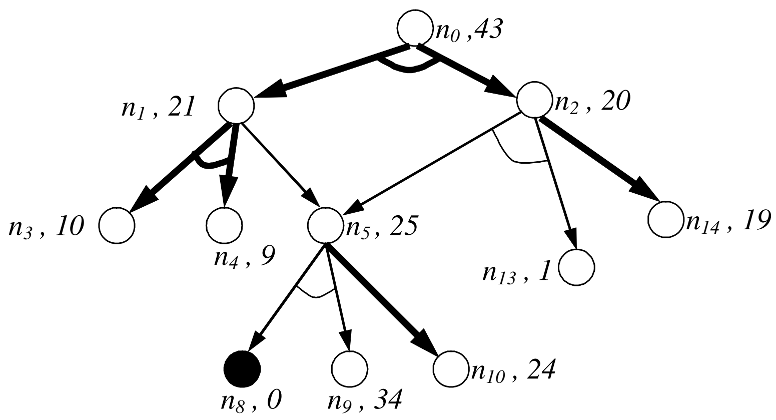
5) $S = \{ n_0, n_2\}$

6) No podemos sacar n_0 de S, ya que n_2 es descendiente suyo y ya está en S. Por tanto, sacamos n_2 de S. El nuevo coste de n_2 es 20, ya que ha de redirigirse el enlace que apunta al mejor subgrafo que cuelga de n_2. Por tanto, habría que introducir su padre, n_0, en S. Esto último no es necesario, ya que n_0 ya estaba incluido en S.

7) Se saca n_0 de S. Su nuevo coste es 43.

8) $S = \{\}$

> La situación final es la siguiente:



- **Ciclo 5:**

> Se expande n_3, aunque también se podría haber expandido n_4 o n_{14}.

> La evolución del conjunto S es la siguiente:

1) $S = \{n_3\}$

2) Se saca n_3 de S. Su nuevo coste es 5. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_2 en S.

3) $S = \{n_1\}$

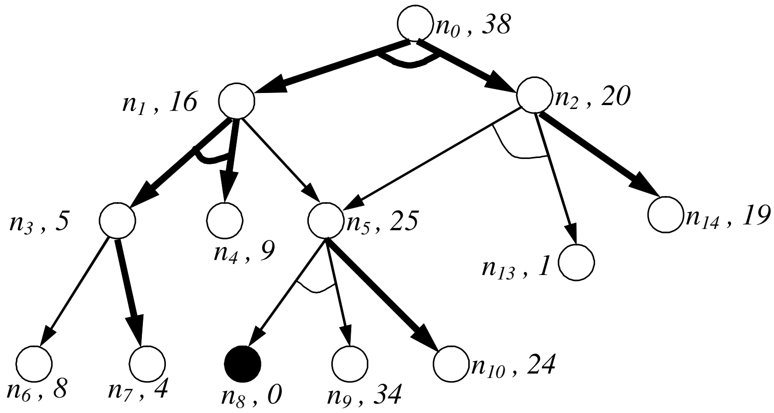
4) Se saca n_1 de S. Su nuevo coste es 16. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_1 en S.

5) $S = \{n_0\}$

6) Se saca n_0 de S. Su nuevo coste es 38.

7) $S = \{\}$

> La situación final es la siguiente:



- **Ciclo 6:**

  > Se expande n_4, aunque también se podría haber expandido n_7 o n_{14}.

  > La evolución del conjunto S es la siguiente:

  1) $S = \{n_4\}$

  2) Se saca n_4 de S. Su nuevo coste es 5. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_4 en S.

  3) $S = \{n_1\}$

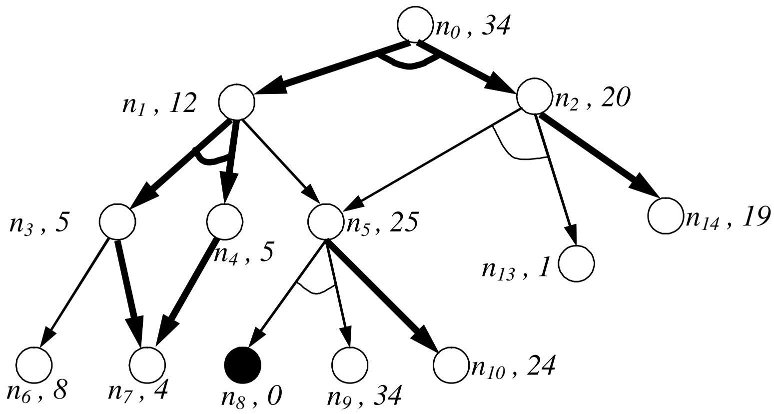
  4) Se saca n_1 de S. Su nuevo coste es 12. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_1 en S.

  5) $S = \{n_0\}$

  6) Se saca n_0 de S. Su nuevo coste es 34.

  7) $S = \{\}$

  > La situación final es la siguiente:



- **Ciclo 7:**

  > Se expande n_7, aunque también se podría haber expandido n_{14}.

  > La evolución del conjunto S es la siguiente:

  1) $S = \{n_7\}$

  2) Se saca n_7 de S. Su nuevo coste es 1, quedando resuelto. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_7 en S.

  3) $S = \{n_3, n_4\}$

  4) Se saca n_3 de S, aunque también se podría haber sacado n_4 ya que ninguno de los dos nodos tiene descendientes en S. Su nuevo coste es 2, quedando resuelto.

Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_3 en S.

5) $S = \{n_1, n_4\}$

6) Se saca n_4 de S, ya que n_1 no se puede sacar debido al hecho de que n_4 es descendiente de n_1 y está contenido en S. El nuevo coste de n_4 es 2, quedando resuelto.

7) $S = \{n_1\}$

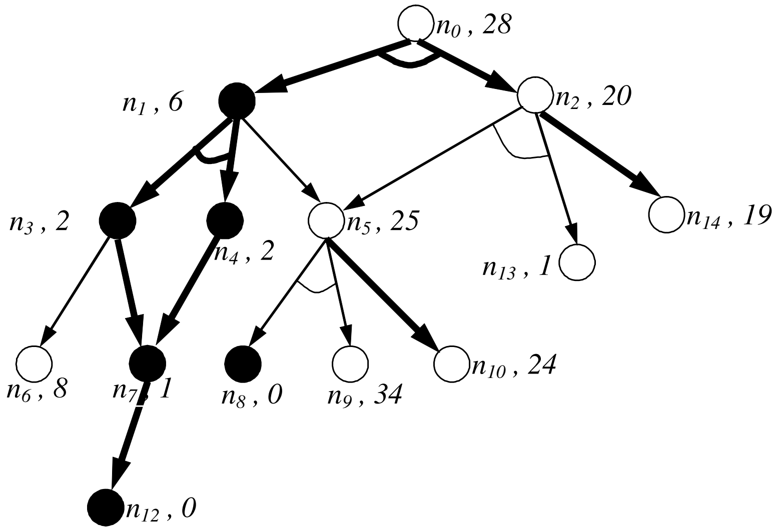
8) Se saca n_1 de S. Su nuevo coste es 6, quedando resuelto. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_1 en S.

9) $S = \{n_0\}$

10) Se saca n_0 de S. Su nuevo coste es 28.

11) $S = \{\}$

> La situación final es la siguiente:



- **Ciclo 8:**

> Se expande n_{14}.

> La evolución del conjunto S es la siguiente:

1) $S = \{n_{14}\}$

2) Se saca n_{14} de S. Su nuevo coste es 1, quedando resuelto. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_{14} en S.

3) $S = \{n_2\}$

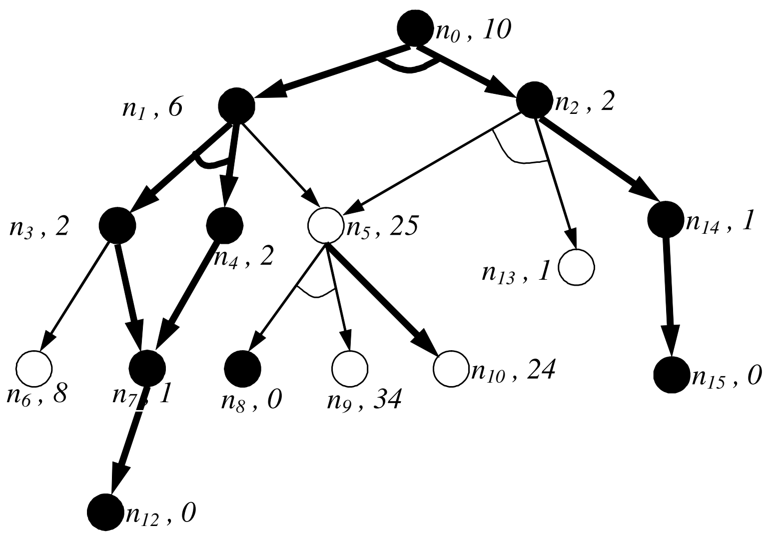
4) Se saca n_2 de S. Su nuevo coste es 2, quedando resuelto. Al haber cambiado este coste respecto al antiguo, se introducen los padres de n_2 en S.

5) $S = \{n_0\}$

6) Se saca n_0 de S. Su nuevo coste es 10.

7) $S = \{\}$

> La situación final es la siguiente:



Al final de este ciclo hemos encontrado un subgrafo solución para el nodo n_0, por lo que el algoritmo finaliza. Obsérvese que el subgrafo solución hallado tiene coste 10 a pesar de estar formado por 9 arcos. Ello es debido a que el arco (n_7, n_{12}) se cuenta dos veces, una para hallar el coste de n_3 y otra para hallar el coste de n_4, a la hora de calcular el coste del subgrafo solución.

Ejercicio 3. (Valoración: 10 / 3)
Explique detalladamente cómo se puede representar mediante Lógica de Predicados clásica el conocimiento de una red de marcos, según la versión más actual de los mismos. Ponga un ejemplo de lo anterior perteneciente al mundo real e ilustre mediante el mismo las deficiencias que presenta la representación de marcos mediante Lógica de Predicados clásica.

SOLUCIÓN

La versión más actual de marcos maneja dos tipos de **nodos**: clases e instancias. Cada nodo consta de una serie de campos de manera que cada campo toma un valor. Por otra parte, la versión más actual de marcos incluye dos tipos de **enlaces**: "instancia-de" y "subclase-de". El primero de ellos une una instancia con la clase a la que pertenece, mientras que el segundo une una subclase con la clase en la que está incluida.

De forma general, denotemos por:

Clase#1: una determinada clase

Campo#1: un campo de la clase Clase#1

Valor#1: el valor que toma el campo Campo#1 de la clase Clase#1

Clase#2: una subclase de la clase Clase#1

Campo#2: un campo de la clase Clase#2

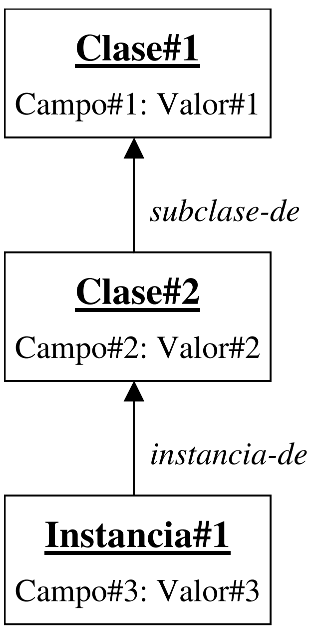
Valor#2: el valor que toma el campo Campo#2 de la clase Clase#2

Instancia#1: una instancia de la clase Clase#2

Campo#3: un campo de la instancia Instancia#1

Valor#3: el valor que toma el campo Campo#3 de la instancia Instancia#1

La red de marcos genérica correspondiente viene representada en la siguiente figura:



En Lógica de Predicados, el contenido de la red de marcos anterior sería la siguiente:

1) Fórmulas asociadas a Instancia#1:

Clase#2(Instancia#1)

Campo#3(Instancia#1, Valor#3)

2) Fórmulas asociadas a Clase#2:

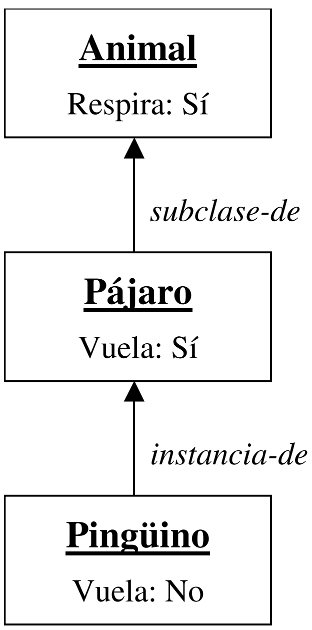
$\forall x$ Clase#2(x) $\rightarrow$ Campo#2(x, Valor#2)

$\forall x$ Clase#2(x) $\rightarrow$ Clase#1(x)

3) Fórmulas asociadas a Clase#1:

$\forall x$ Clase#1(x) $\rightarrow$ Campo#1(x, Valor#1)

Cualquier otra red de marcos con un número cualquiera de clases e instancias se representaría en Lógica de Predicados clásica de forma análoga a la expresada más arriba. Un ejemplo tomado del mundo real sería el siguiente:



El contenido de la red anterior en Lógica de Predicados sería:

Pájaro(Pingüino)	(1)
Vuela(Pingüino, No)	(2)
$\forall x$ Pájaro(x) $\rightarrow$ Vuela(x, Sí)	(3)
$\forall x$ Pájaro(x) $\rightarrow$ Animal(x)	
$\forall x$ Animal(x) $\rightarrow$ Respira(x, Sí)	

Los problemas asociados a la representación en Lógica de Predicados clásica provienen del hecho de no poder modelar la no monotonía propia de los marcos. Por ejemplo, en la red del mundo real dibujada anteriormente, aunque sabemos que un pájaro vuela, sólo afirmaríamos que un pingüino vuela si no hay información más específica que indique lo contrario. Como esta información existe, se afirmará a partir de la red de marcos que un pingüino no vuela. Por el contrario, se puede deducir de la representación lógica de la red de marcos del mundo real que Vuela(Pingüino, No), a partir de (2), y que Vuela(Pingüino, Sí), a partir de (1) y (3), sin que sea posible deshacer esta inconsistencia.