

Practica 3

Búsqueda en juegos



Adrián Rodríguez Escudero
SISTEMAS INTELIGENTES | GRUPO DE TRABAJO 2

Enunciado

El objetivo de estas sesiones de trabajo es introducir el lenguaje PROLOG y utilizar la lógica de predicados como herramienta para la representación del conocimiento y construcción de sistemas basados en reglas. La tabla siguiente especifica el trabajo que el alumno debe realizar tanto durante la sesión presencial de grupo de trabajo, como de forma autónoma.

Búsqueda en juegos	
Sesión de trabajo	<ul style="list-style-type: none"> Familiarícese con el entorno de desarrollo en PROLOG, estudiando los apartados 1 a 3 del guion y probando los ejemplos. Resuelva la relación "Ejercicios Prolog – 1ª Sesión de trabajo". Siga las instrucciones y no olvide adjuntar los resultados a la Tarea 03 – Razonamiento basado en conocimiento: PROLOG". 1. Representa en Prolog el árbol de familia siguiente: <div data-bbox="710 772 1300 996" data-label="Diagram"> <pre> graph TD CJ[Carmen & Javier] --- AO[Ana & Óscar] CJ --- MF[María & Francisco] AO --- EEI[Enrique, Eva, Isabel] MF --- A[Antonio] </pre> </div> <p>Para ello utilice los predicados hijo y descendiente. Realice las consultas siguientes y explique las modificaciones que realiza sobre la representación inicial</p> <ol style="list-style-type: none"> ¿Es Eva hija de Óscar y Ana? ¿Es Eva descendiente de Ana? ¿Es Eva descendiente de Javier? ¿Es Antonio descendiente de Carmen? 2. Cree un programa PROLOG para resolver el problema del 8-puzzle utilizando como base la implementación del problema del granjero, el lobo, la oveja y la col.
Trabajo autónomo	<ol style="list-style-type: none"> 3. Un robot aspiradora se desenvuelve en un espacio con dos ubicaciones en las que puede habersudado no. El robot siempre se encuentra en una de las dos ubicaciones y puede realizar tres acciones: desplazarse a la izquierda, desplazarse a la derecha y aspirar toda la suciedad. Cree un programa en PROLOG para que el robot aspire toda la suciedad utilizando como base la implementación del problema del granjero, el lobo, la oveja y la col.. 4. Dada la base de conocimiento formada por el conjunto de reglas siguiente: <ul style="list-style-type: none"> R1: Si A∧B, entonces C R2: Si D∧E∧F, entonces G R3: Si H∧I, entonces J R4: Si C∧G, entonces K R5: Si G∧J, entonces L R6: Si K∧L, entonces M Los hechos disponibles son $G = v$ y $L = f$. Si tomamos J como variable objetivo, ¿qué conclusión se obtiene en Prolog? Implemente el sistema basado en reglas utilizando encadenamiento hacia delante (SBR_adelante.pl) y hacia atrás (SBR_atras.pl). Incluya en cada uno de ellos como comentarios la consulta que utiliza y el resultado de la misma.

Ejercicio 1

4 ?- hija(eva, ana), hija(eva, oscar) .

5 ?- descendiente(eva, ana) .

6 ?- descendiente(eva, javier) .
false.

7 ?- descendiente(antonio, carmen) .
false.

Ejercicio 2

4 ?- plan(Objetivo, Objetivo, Camino, Camino) .

5 ?- plan(Camino, Camino, Camino, Camino) .

6 ?- plan(Objetivo, Camino, Camino, Camino) .
Objetivo = Camino ;
false.

7 ?- plan(Objetivo, Objetivo, Objetivo, Camino) .
Objetivo = Camino ;
false.

8 ?- plan(Objetivo, Objetivo, Objetivo, Objetivo) .

9 ?- accion(estado(cinco, seis, siete, cuatro, cero, ocho, tres, dos, uno),
estado(uno, dos, tres, ocho, cero, cuatro, siete, seis, cinco)) .
false.

Ejercicio 3

```
4 ?- limpiar_izquierda(izquierda,sucio,limpio).  
5 ?- limpiar_izquierda(izquierda,limpio,limpio).  
false.  
6 ?- limpiar_izquierda(izquierda,limpio,sucio).  
false.  
7 ?- limpiar_izquierda(izquierda,sucio,sucio).  
false.  
8 ?- limpiar_izquierda(derecha,sucio,limpio).  
false.  
9 ?- limpiar_izquierda(derecha,limpio,limpio).  
false.  
10 ?- limpiar_izquierda(derecha,limpio,sucio).  
false.  
11 ?- limpiar_izquierda(derecha,sucio,sucio).  
false.  
12 ?- desplazar(izquierda,derecha).  
13 ?- desplazar(izquierda,izquierda).  
false.  
14 ?- desplazar(derecha,izquierda).  
15 ?- desplazar(derecha,derecha).  
false.
```

Ejercicio 4

```
4 ?- fact(a).  
false.  
  
5 ?- fact(b).  
false.  
  
6 ?- fact(c).  
  
7 ?- fact(d).  
false.  
  
8 ?- fact(e).  
false.  
  
9 ?- fact(f).  
false.  
  
10 ?- fact(g).  
  
11 ?- fact(h).  
false.  
  
12 ?- fact(i).  
false.  
  
13 ?- fact(j).  
false.  
  
14 ?- fact(k).  
false.  
  
15 ?- fact(k).  
false.  
  
16 ?- fact(l).  
false.  
  
17 ?- fact(m).  
false.
```



La conclusión a la que llega el programa es que todos los hechos son falsos excepto las que hemos señalado como verdaderos, con lo cual, g y c son los únicos hechos verdaderos.