

## EXAMEN DE META-INTÉRPRETE DE PROLOG

Al dorso se encuentra el esquema visto en clase de la instalación eléctrica de una casa con el que realizar una base de conocimiento para hallar una solución a los siguientes ejercicios, los cuales no son más que una mera modificaciones del correspondiente meta-intérprete Vanilla:

```
:-op(40, xfy, &).  
:-op(50, xfy, --->).
```

```
solve(true):- !.  
solve((A,B)) :- !, solve(A), solve(B).  
solve(A) :- clause(A, B), solve(B).
```

1. Modificar dicho meta-intérprete para que resuelva la consulta dada (A) bajo un nivel de profundidad máximo (Prf) pasado como segundo argumento: (3 ptos).

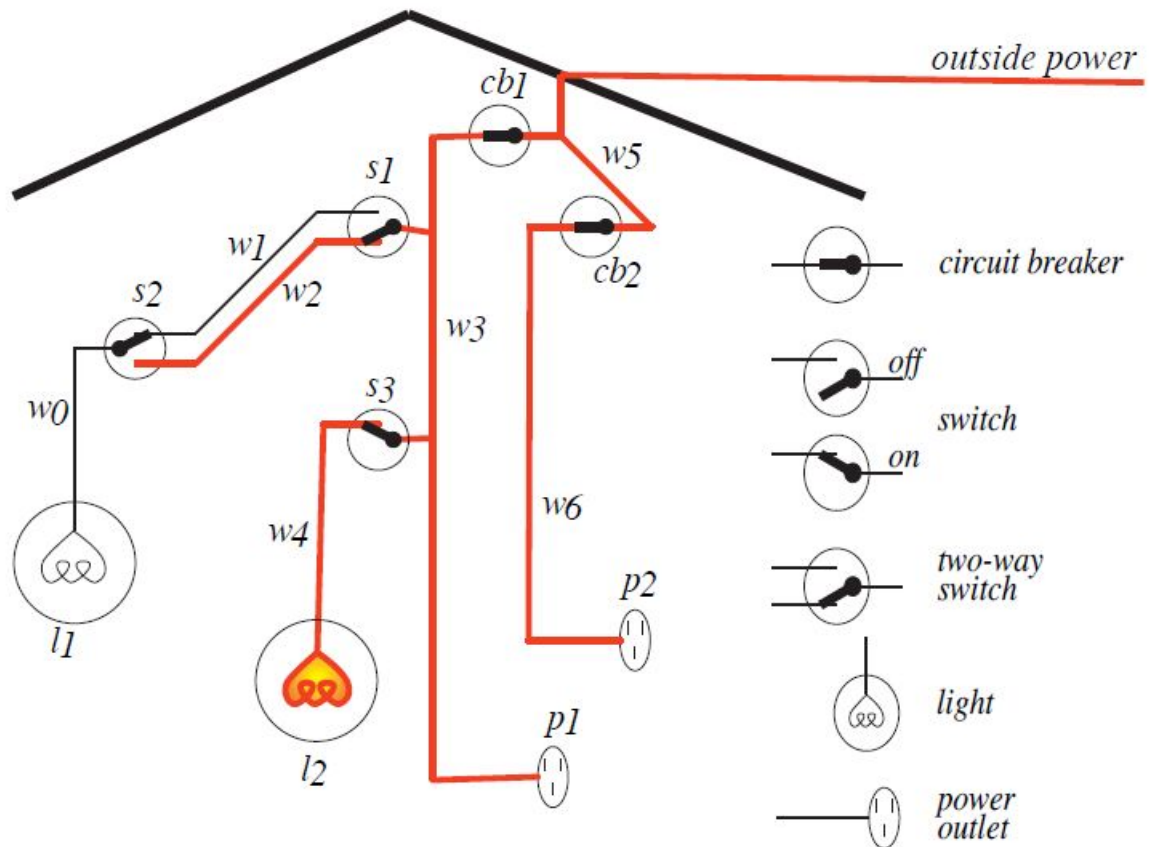
```
solve_pmax(A, Prf)
```

2. Cambiar la solución anterior, para que el meta-intérprete evalúe derecha a izquierda. (2 ptos).
3. Construir un meta-intérprete basado en Vanilla, para que construya una especie de árbol de pruebas. Así, cuando se pregunta:

```
solve(A, (A:-C))
```

va ir sacando la lista de reglas que va aplicando para resolver A. Se sugiere que en Vanilla, se use una variable auxiliar cada vez que se llame (a sí mismo). Ver ejemplo al dorso: (5 ptos).

**ENTREGA:** deberá crear un fichero con la solución de cada ejercicio, en el que a modo de comentario figuren su nombre y dos apellidos. A continuación, serán comprimidos en un solo fichero ZIP, cuyo nombre deberá pedir al profesor vigilante del laboratorio justo antes de subir dicho fichero a la plataforma moodle de la asignatura.



### Ejemplo 3:

BASE DE CONOCIMIENTO:

valor(w1, 1).

conectado(w2, w1).

conectado(w3, w2).

valor(W,X):-conectado(W,V), valor(V,X).

CONSULTA:

solve\_proof(valor(W,V), (valor(W,V):-B)).

W = w1,

X = 1,

B = true ;

W = w2,

X = 1,

B = ((conectado(w2, w1):-true), (valor(w1, 1):-true)) ;

W = w3,

X = 1,

B = ((conectado(w3, w2):-true), (valor(w2, 1):- (conectado(w2, w1):-true), (valor(w1, 1):-true))) ;