

## PRÁCTICA 2: PROGRAMACIÓN EN ISO PROLOG

**Profesor:** Miguel García Remesal

### ENUNCIADO DEL SUPUESTO PRÁCTICO

Una puerta esta gobernada por una serie de palancas, cada una de las cuales puede estar en una de dos posibles posiciones (on/off), de manera que la puerta se abre solo con una configuración determinada de posiciones de las palancas y permanece cerrada con el resto. Se conoce cual es la configuración que hace abrirse la puerta, pero se desconoce como llegar a ella a partir de una configuración cualquiera dada inicialmente.



Cada palanca se puede mover de su posición actual a la otra posición, pero, al hacerlo, el mecanismo interno de la puerta automáticamente cambia de posición a algunas otras palancas (o a ninguna). Se conoce cuales de las palancas cambian de posición al mover cada una de ellas. Se dispone de esta información en una lista de pares de la forma P-L, donde P es el identificador de una palanca y L una lista (no vacía) de los identificadores de las palancas que cambian automáticamente al mover la palanca P. Las palancas se identifican con números del 1 en adelante.



El trabajo propuesto en relación a este supuesto consistiría en programar un predicado `abrir/4`, cuya cabecera sería `abrir(Estado_Inicial, Estado_Final, Movimientos_Automaticos, Solucion)`, siendo `Movimientos_Automaticos` la lista de cambios automáticos de posición de las palancas, tal como se ha definido mas arriba, `Solucion` la lista de identificadores de las palancas que hay que mover para llegar a la configuración `Estado_Final`, que hace abrirse la puerta a partir de la configuración de partida `Estado_Inicial`, en el orden adecuado en que hay que mover las palancas para conseguirlo. Las configuraciones de posiciones de  $n$  palancas se representan con términos `palancas/n` en los que cada uno de los  $n$  argumentos es `on` u `off` — por ejemplo, `palancas(on,off,on)` — determinando la posición de la palanca identificada por el número del correspondiente argumento.

### SE PIDE:

1. **(8 puntos)** Escribe un programa para el predicado **`abrir/4`** descrito mas arriba. Este predicado solo es necesario que funcione con los tres primeros argumentos instanciados. Aun así, una instancia particular del problema (un conjunto concreto de valores de esos tres argumentos) puede tener distintas soluciones y el predicado debe devolver, por vuelta atrás, todas ellas. El programa no debe entrar en bucle infinito, ni siquiera por vuelta atrás. **Nota:** Si una instancia concreta del problema tiene infinitas soluciones, el predicado `abrir/4` debe devolver infinitas respuestas por vuelta atrás. Esto **no** se considera bucle infinito.
2. **(2 puntos)** Puede ser que el programa del punto anterior no devuelva en primer lugar la(s) solución (soluciones) mas corta(s), que sería lo deseable. Escribe un predicado **`minimo_abrir/4`** (con los mismos argumentos y en el mismo orden que **`abrir/4`**) que utilice el programa anterior para obtener la solución más corta posible para el problema de la puerta (nótese que puede haber varias soluciones mínimas). Asegúrate de que devuelve, por vuelta atrás, todas las soluciones mínimas.

### CASOS DE PRUEBA DE EJEMPLO

- `abrir (palancas(on,on,on),palancas(on,on,on),[], Solucion).`

- abrir(palancas(on,on,on), palancas(off,off,off),[],Solucion).
- abrir(palancas(on,on,on), palancas(off,off,off), [1-[2,3],2-[1,3],3-[2,1]],Solucion).
- abrir(palancas(on,on,on,on,on,on),palancas(off,off,off,off,off,off),[1-[3,4],6-[3,4],7-[4,5],2-[4,5],3-[1,6],5-[2,7],4-[1,2,6,7]],Solucion).

## **NORMAS GENERALES DE PRESENTACIÓN Y RESTRICCIONES**

1. La solución a ambos apartados debe entregarse en un fichero denominado **palancas.pl**.
2. Debe entregarse una memoria escrita donde se explique claramente el significado y el funcionamiento de cada predicado definido en el fichero palancas.pl. La memoria debe redactarse en un fichero de texto plano denominado **memoria.txt**.
4. Los dos ficheros anteriores deben subirse a la plataforma Moodle por separado (no empaquetados).
5. Esta práctica versa sobre programas en ISO Prolog, y por tanto, es posible utilizar cualquier construcción contemplada en dicho lenguaje de programación y que haya sido vista en clases de teoría.
8. El no respetar estas normas implica suspender la práctica, dado que un corrector automático hará un primer «screening» de las mismas para comprobar su corrección. Si alguna de estas convenciones no se respeta, el programa corrector rechazará la práctica sin corregirla, y por tanto se aplicará la regla:

suspenso(Alumno) :- rechazada\_por\_corrector(practica(Alumno)).

Por tanto, el alumno debe ser extremadamente cuidadoso a la hora de revisar que se cumplen todos los requisitos reseñados más arriba antes de entregar la práctica.

## **CALIFICACIONES**

La práctica se evaluará de 0 a 10 puntos distribuidos como se especifica en el enunciado.

## **REALIZACIÓN DE LA PRÁCTICA POR GRUPOS**

La práctica puede realizarse en grupos de 4 personas como máximo. Es posible crear grupos que involucren personas de la misma titulación siempre que asistan al mismo grupo de teoría. El profesor no facilitará la búsqueda de grupo a ningún alumno, dado que siempre existe la opción de realizar la práctica en solitario.

## **FECHA DE ENTREGA**

La fecha tope de entrega es el Jueves 19 de diciembre de 2013 a las 17:00 horas. No se admitirá ninguna práctica entregada fuera de fecha. No olvidar CONFIRMAR las entregas en Moodle. No se corregirán borradores.

## **POLÍTICA DEL PROFESORADO ANTE CASOS DE PLAGIO**

El plagio es un tema muy serio que el profesorado de esta asignatura califica como una falta muy grave. Las prácticas entregadas por el alumnado serán sometidas a un proceso de «screening» para detectar irregularidades. En caso de detectarse alguna irregularidad, el (los) alumno(s) responsables (plagiador y plagiado) serán sancionados, y perderán todas las notas

que hubieran obtenido con anterioridad, y serán evaluados como suspenso en todas las partes de la asignatura hasta la misma convocatoria del curso académico siguiente (excluida). A estos efectos, todos los alumnos miembros de un grupo son corresponsables. Es responsabilidad de cada alumno/grupo la protección de su propia información.