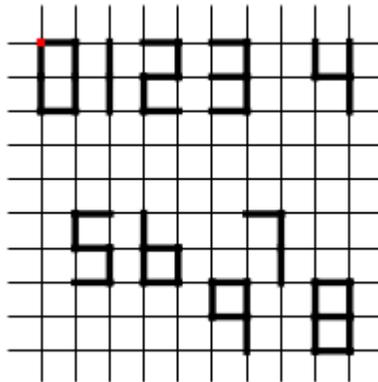


Práctica 1 (Programación Lógica Pura): Reconocimiento automático de dígitos

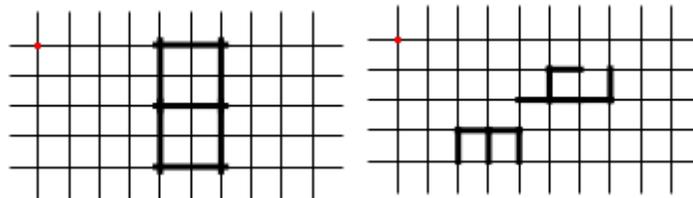
Profesor responsable: Miguel García Remesal

Se dispone de una retícula infinita con origen en la esquina superior izquierda ($X=0, Y=0$) y ejes de abscisas y ordenadas crecientes hacia la derecha y hacia abajo respectivamente (es decir, no hay coordenadas con abscisas y/u ordenadas negativas). En esta retícula hay definidos un número finito de segmentos, que están representados en la base de hechos del programa mediante el predicado `lit/4`. Este predicado, cuya cabecera es `lit(X1, Y1, X2, Y2)`, será cierto si existe en la retícula un segmento definido mediante las coordenadas $(X1, Y1)$ y $(X2, Y2)$ expresadas mediante la notación de Peano. Por ejemplo, el hecho `lit(s(s(0)), s(0), 0, s(s(s(0))))` indica la existencia de un segmento delimitado por los puntos $(2, 1)$ y $(0, 3)$.

Al igual que ocurre con los displays digitales de siete segmentos, los segmentos pueden utilizarse para formar dígitos decimales. Por ejemplo, en la figura 1a puede observarse la composición de cada uno de los dígitos del 0 al 9.



(a) Todos los dígitos decimales.



(b) Esto NO es un ocho.

(c) Dígitos rotados y ocultos.

Figura 1: ejemplos de retículas con dígitos.

El objetivo de este trabajo práctico es escribir un predicado `number_in_grid/1`

que devuelva por vuelta hacia atrás todos los dígitos que son reconocibles a partir de la retícula definida por los hechos presentes en la base de hechos del programa. El predicado `number_in_grid/1` no debe reconocer dígitos a escalas diferentes de la que se muestra en la figura 1a (por ejemplo, la forma que aparece en la figura 1b no debe reconocerse como el dígito 8). Además, la orientación de los dígitos no tiene porqué ser la misma que la que aparece en la figura 1a, sino que estos pueden aparecer rotados. Asimismo, un mismo segmento puede utilizarse para reconocer más de un dígito (por ejemplo, los dígitos 4 y 1 están “escondidos” dentro de un 9, que a su vez puede reconocerse como un 6 rotado 180° en la dirección de las agujas del reloj). La figura 1c muestra un ejemplo de esta casuística, dado que en este caso el programa debe devolver los dígitos 1, 3, 4 y 7 como soluciones válidas al problema planteado.

Normas

La fecha límite de entrega de la práctica es el **Jueves 5 de Noviembre de 2015 a las 12:00 horas**. Esta práctica puede hacerse en grupos de 3 personas. El código debe entregarse en un fichero denominado EXACTAMENTE `number.pl` que implemente el predicado pedido en el enunciado. Dado que esta práctica será corregida utilizando un corrector automático, es OBLIGATORIO que el functor, aridad y orden de los argumentos DE TODOS LOS PREDICADOS mencionados en el enunciado sea EXACTAMENTE IGUAL al indicado en el enunciado. Si esto no se hace así, el corrector automático rechazará la práctica y se calificará como suspenso.

Dado que estas instrucciones están CLARAMENTE EXPLICADAS en las normas, no se aceptarán prácticas que no cumplan dichas especificaciones. Es responsabilidad de(los) alumnos el asegurarse bien de que se cumplan las especificaciones antes de efectuar la entrega. Se recuerda al alumno que no es posible utilizar ningún recurso de ISO Prolog ni ningún predicado predefinido. El uso de este tipo de recursos conlleva un decremento sustancial de la nota o incluso el suspenso de la misma dependiendo de los recursos utilizados.

Además del fichero `number.pl`, deberá entregarse otro fichero denominado `memoria.txt` donde se explique brevemente el desarrollo de la práctica. Ambos ficheros deben comprimirse en un único fichero ZIP, RAR o TAR que tenga por nombre el número de matrícula del alumno y debe subirse al Moodle de manera individual. Es decir, todos los alumnos del grupo deben subir el fichero al Moodle. En el fichero `memoria.txt` debe aparecer el nombre y número de matrícula de todos y cada uno de los alumnos pertenecientes al grupo.