PRÁCTICA 2: PROGRAMACIÓN ISO-PROLOG

(Curso 2017-2018)

Una fábrica metalúrgica se está haciendo famosa gracias a unas cadenas que produce que han sido diseñadas por ella y resultan especialmente resistentes. Una de las claves del éxito está en los eslabones que emplean, que tienen un cierre especial en cada uno de sus dos extremos. Han diseñado cierres de tipos muy distintos, de forma que dos eslabones se pueden ensamblar entre sí por sus extremos solo si ambos extremos tienen el mismo tipo de cierre.

El departamento de calidad está sometiendo las cadenas a pruebas de resistencia, para lo que utiliza cadenas cerradas, esto es, con eslabones ensamblados en línea uno tras otro y el último ensamblado con el primero. Para ayudarse en sus pruebas han solicitado un programa informático con las siguientes características. La realización del correspondiente código Prolog es lo que se solicita a los alumnos en esta práctica.

El predicado **cierre/2** debe tener éxito cuando su primer argumento es una lista de eslabones sin repeticiones y el segundo una lista que representa una cadena cerrada de eslabones de la primera lista (todos o solo algunos). Solo es necesario que funcione correctamente cuando el primer argumento se da en la llamada.

Los eslabones se representan con términos eslabon/2 cuyos argumentos representan los tipos de cierre en cada uno de sus extremos. Un eslabón con cierres tipo a y b se puede representar como eslabon(a,b) o eslabon(b,a) pero las listas de eslabones (que no tienen repeticiones) solo contendrán uno de ellos. Los tipos de cierres de un mismo eslabón nunca son iguales. Una lista de eslabones que representa una cadena cerrada debe contenerlos en el orden en que ensamblan y el último eslabón debe poder ensamblar con el primero.

Desgraciadamente, una cadena cerrada se puede representar como lista de eslabones de muchas formas distintas, dependiendo del que se coloque como primer elemento: [A,B,C,D], [B,C,D,A], [C,D,A,B] y [D,A,B,C] representarían la misma cadena cerrada. Esto también ocurre si la cadena se recorre en un sentido o en el otro: [D,C,B,A] también representa la misma cadena cerrada. Algo similar puede ocurrir por otras razones, pero en todos los casos en que ocurre la diferencia entre las listas está en el orden, no en los eslabones que contienen.

Como lo que interesa de las cadenas cerradas para las pruebas de resistencia son los eslabones que la forman, no el orden en que se ensamblan, esa multiplicidad de listas que representan la misma cadena resulta excesiva. Así que de todas las listas de cadenas cerradas que contengan los mismos elementos pero en distinto orden basta con obtener solo una de ellas. También va a interesar obtener el número mínimo de eslabones necesarios para formar una cadena cerrada.

El predicado **cierreUnico/2** debe comportarse igual que cierre/2 pero de forma que cada solución que devuelva en su segundo argumento no debe contener los mismos elementos que otra de las soluciones que devuelve. Es decir, en el caso [A,B,C,D] del ejemplo anterior solo debe devolver una de las (ocho) listas posibles. En cambio, el predicado cierre/2 debe devolver siempre todas, sin importar que solo difieran en el orden.

El predicado **cierreMinimo/2** debe tener éxito cuando su primer argumento es una lista de eslabones sin repeticiones y su segundo argumento es el mínimo número de eslabones que contiene que pueden formar cadena cerrada. En este predicado se valorará positivamente que el código no obtenga todas las soluciones posibles para luego compararlas.

Cuando se ensamblan físicamente dos eslabones se orientan de manera que los extremos de cada uno que se van a unir estén enfrentados el uno al otro. Sin embargo, esto no ocurre en el programa. Si la lista inicial de eslabones contiene eslabon(a,b), éste se puede utilizar para una cadena cerrada enlazándolo mediante a con el anterior y mediante b con el siguiente, pero también al revés: mediante b con el anterior y mediante a con el siguiente, aunque el término que aparezca en ambos casos en la lista de la cadena cerrada debe ser eslabon(a,b). (Es decir, no se "re-orienta" poniéndolo como eslabon(b,a)).

Ejemplos

?- cierreMinimo([eslabon(c,b),eslabon(c,d),eslabon(a,b),eslabon(e,b),eslabon(d,b),eslabon(a,e)], Min).

Min = 3 ? ;

no

?- cierreUnico([eslabon(c,b),eslabon(c,d),eslabon(a,b),eslabon(e,b),eslabon(d,b),eslabon(a,e)], Cierre).

Cierre = [eslabon(a,b),eslabon(a,e),eslabon(e,b)] ? ;

Cierre = [eslabon(c,b),eslabon(c,d),eslabon(d,b)] ? ;

Cierre = [eslabon(c,b),eslabon(c,d),eslabon(d,b),eslabon(a,b),eslabon(a,e),eslabon(e,b)] ? ;

no

?- cierre([eslabon(c,b),eslabon(c,d),eslabon(a,b),eslabon(e,b),eslabon(d,b),eslabon(a,e)], Cierre).

Cierre = [eslabon(a,b),eslabon(a,e),eslabon(e,b)] ? ;

Cierre = [eslabon(a,b),eslabon(a,e),eslabon(e,b),eslabon(c,b),eslabon(c,d),eslabon(d,b)] ? ;

Cierre = [eslabon(a,b),eslabon(a,e),eslabon(e,b),eslabon(d,b),eslabon(c,d),eslabon(c,b)] ? ;

Cierre = [eslabon(a,b),eslabon(c,b),eslabon(c,d),eslabon(d,b),eslabon(e,b),eslabon(a,e)] ? ;

Cierre = [eslabon(a,b),eslabon(e,b),eslabon(a,e)] ? ;

Cierre = [eslabon(a,b),eslabon(d,b),eslabon(c,d),eslabon(c,b),eslabon(e,b),eslabon(a,e)] ? ;

Cierre = [eslabon(a,e),eslabon(a,b),eslabon(c,b),eslabon(c,d),eslabon(d,b),eslabon(e,b)] ? ;

Cierre = [eslabon(a,e),eslabon(a,b),eslabon(e,b)] ? ;

Cierre = [eslabon(a,e),eslabon(a,b),eslabon(d,b),eslabon(c,d),eslabon(c,b),eslabon(e,b)] ? ;

Cierre = [eslabon(a,e),eslabon(e,b),eslabon(c,b),eslabon(c,d),eslabon(d,b),eslabon(a,b)] ? ;

Cierre = [eslabon(a,e),eslabon(e,b),eslabon(a,b)] ? ;

Cierre = [eslabon(a,e),eslabon(e,b),eslabon(d,b),eslabon(c,d),eslabon(c,b),eslabon(a,b)] ? ;

Cierre = [eslabon(c,b),eslabon(c,d),eslabon(d,b)] ? ;

Cierre = [eslabon(c,b),eslabon(c,d),eslabon(d,b),eslabon(a,b),eslabon(a,e),eslabon(e,b)] ? ;

Cierre = [eslabon(c,b),eslabon(c,d),eslabon(d,b),eslabon(e,b),eslabon(a,e),eslabon(a,b)] ? ;

Cierre = [eslabon(c,b),eslabon(a,b),eslabon(a,e),eslabon(e,b),eslabon(d,b),eslabon(c,d)] ? ;

Cierre = [eslabon(c,b),eslabon(e,b),eslabon(a,e),eslabon(a,b),eslabon(d,b),eslabon(c,d)] ? ;

Cierre = [eslabon(c,b),eslabon(d,b),eslabon(c,d)] ? ;

Cierre = [eslabon(c,d),eslabon(c,b),eslabon(a,b),eslabon(a,e),eslabon(e,b),eslabon(d,b)] ? ;

Cierre = [eslabon(c,d),eslabon(c,b),eslabon(e,b),eslabon(a,e),eslabon(a,b),eslabon(d,b)] ? ;

Cierre = [eslabon(c,d),eslabon(c,b),eslabon(d,b)] ? ;

Cierre = [eslabon(c,d),eslabon(d,b),eslabon(c,b)] ? ;

Cierre = [eslabon(c,d),eslabon(d,b),eslabon(a,b),eslabon(a,e),eslabon(e,b),eslabon(c,b)] ? ;

Cierre = [eslabon(c,d),eslabon(d,b),eslabon(e,b),eslabon(a,e),eslabon(a,b),eslabon(c,b)] ? ;

Cierre = [eslabon(d,b),eslabon(c,d),eslabon(c,b)] ? ;

Cierre = [eslabon(d,b),eslabon(c,d),eslabon(c,b),eslabon(a,b),eslabon(a,e),eslabon(e,b)] ? ;

Cierre = [eslabon(d,b),eslabon(c,d),eslabon(c,b),eslabon(e,b),eslabon(a,e),eslabon(a,b)] ? ;

Cierre = [eslabon(d,b),eslabon(c,b),eslabon(c,d)] ? ;

Cierre = [eslabon(d,b),eslabon(a,b),eslabon(a,e),eslabon(e,b),eslabon(c,b),eslabon(c,d)] ? ;

Cierre = [eslabon(d,b),eslabon(e,b),eslabon(a,e),eslabon(a,b),eslabon(c,b),eslabon(c,d)] ? ;

Cierre = [eslabon(e,b),eslabon(a,e),eslabon(a,b)] ? ;

Cierre = [eslabon(e,b),eslabon(a,e),eslabon(a,b),eslabon(c,b),eslabon(c,d),eslabon(d,b)] ? ;

Cierre = [eslabon(e,b),eslabon(a,e),eslabon(a,b),eslabon(d,b),eslabon(c,d),eslabon(c,b)] ? ;

Cierre = [eslabon(e,b),eslabon(c,b),eslabon(c,d),eslabon(d,b),eslabon(a,b),eslabon(a,e)] ? ;

Cierre = [eslabon(e,b),eslabon(a,b),eslabon(a,e)] ? ;

Cierre = [eslabon(e,b),eslabon(d,b),eslabon(c,d),eslabon(c,b),eslabon(a,b),eslabon(a,e)] ? ;

no

?-