Ing. de Sistemas e Informática
Matemáticas Discretas
Jorge M. Londoño P.

Taller Prolog 5
Unificación, resolución y retroceso

Se dice que dos expresiones ***unifican*** si existe una particularización de las variables que haga idénticas las dos expresiones. A la asignación de variables que unifica las expresiones se le llama ***unificador***. En la interfaz de consultas, la unificación se representa por el signo "=", asi por ejemplo, estos son algunos ejemplos de unificación de expresiones:

```
% Expresiones que si unifican
?- amigos(a,b) = amigos(X,b).
X = a.

?- amigos(X,b) = amigos(a,Y).
X = a,
Y = b.

?- amigos(X,b) = amigos(Z,Y).
X = Z,
Y = b.

% Expresiones que no unifican
?- amigos(a,X) = amigos(b,Y).
false.

?- amigos(a,b) = enemigos(Y,Z).
false.
```

Por el otro lado, el operador \= se entiende como el *no unifican,* es decir, es verdad si no es posible encontrar una particularización de las variables que haga las dos expresiones iguales. Por ejemplo:

```
?- amigos(a,b) \= amigos(a,c).
true.

?- amigos(a,b) \= amigos(X,c).
true.

?- amigos(a,b) \= amigos(a,Y).
false.
```

Observar que el último caso es falso, pues efectivamente existe un unificador de

las dos expresiones: Y=b.

El proceso de ***resolución*** es el proceso por medio del cual el Prolog determina si es posible encontrar un unificador que haga verdadera la consulta. La resolución se puede dar de dos formas:

- Las expresiones que aparecen en la consulta (separadas por comas) unifican con un hecho de la base de datos. En este caso la unificación es inmediata. Si existen varios hechos que satisfacen la consulta, el Prolog los retorna todos al reintentar la consulta presionando ";".
- Las expresiones de la consulta unifican con reglas de la base de datos. En este caso, se reemplaza el encabezado por las metas y se intentan resolver cada una de las metas.
- En cualquiera de los dos casos, si una de las expresiones o metas falla (es falsa), la consulta retorna falso. Se dice en estos casos que la consulta falla.

Observar que en el proceso de resolución se puede repetir en múltiples ocasiones, por ejemplo cuando para resolver cada meta, se unifica con el encabezado de una regla.

Durante el proceso de resolución puede ocurrir que una de las metas falle. Un caso particular falso de la consulta no necesariamente significa que no sea posible satisfacer la consulta. Por este motivo, el Prolog reintenta todas las posibles unificaciones de la meta que falla. Es más, si no se encuentra ningún unificador, el Prolog regresa a la meta anterior y reintenta con otros posibles unificadores. Solo si despues de haber intentando todos los posibles unificadores de todas las posibles metas no se encuentra un unificador de la consulta, entonces el Prolog retorna el resultado falso. Este proceso de devolverse atrás y reintentar resolver cada una de las metas se llama ***retroceso***.

Función de seguimiento de la ejecución:

Para facilitar la búsqueda de errores durante la ejecución de un programa, el Prolog posee el predicado `trace.`, el cual activa el modo de trazado para hacer seguimiento del programa.

Para dar por terminado el modo de seguimiento se utiliza el predicado `notrace`.

Ejercicios:

Para esta práctica se reutilizará la base de datos de la práctica anterior que contiene:

- Hechos `libroLeido/2, libroIdioma/2`

- Regla `leeIdioma/2, leenElMismoIdioma/2`

1) Definir en la base de datos hechos `generoLibro/2` que relacionen un libro

con su respectivo género (novela, ficción, etc.). Por ejemplo, *cien años de soledad* es una *novela*.

2) Definir la regla `genero_idioma/3` que a partir del título de un libro obtenga el género al que pertenece y el idioma en que esta escrito.
3) Utilizando el modo de trazado del programa realizar la consulta para obtener el género y el idoma de un libro. Registrar la traza de la ejecución en el archivo de consultas. Qué unificadores encuentra la consulta? (Utilizar ; para reintentar la consulta y obtener todos los unificadores). Registrar en el archivo de `<NombreApellido>-<ID>-consultas5.txt` la traza de la ejecución, señalando los unificadores encontrados.
4) Definir la regla igualGeneroIdioma/2 la cual permite determina si dos libros tienen el mismo género e idioma.
5) Realizar la consulta que dado un libro determinado, encuentre todos los libros en la base de datos que tienen el mismo género e idioma. Registrar en el archivo `*-consultas5.txt` la traza de la ejecución tanto para un caso afirmativo (hay varias libros que coinciden en género e idioma), como para el caso negativo (no hay otros libros que coincidan). Indicar cuantos reintentos se hicieron de la consulta.
6) Realizar una consulta con negación de la siguiente forma: Cuáles son los libros que no tienen el mismo género e idioma que un libro determinado? Intentar la consulta de dos formas: 1) Sin particularizar previamente la variable que representa el otro libro, y 2) haciendo la particularización previa de la variable. Utilizar el modo de trazado del programa y registrar en el archivo `*-consultas5.txt` como fue la ejecución tanto para cada uno de los casos. Observar en la traza porque falla el primer caso y no el segundo. Indicar si hubo retrocesos durante la ejecución de la consulta.

Informe:

Enviar la base de datos (`<NombreApellido>-<ID>-bd5.pl`) y el informe incluyendo todas las consultas solicitadas (`<NombreApellido>-<ID>-consultas5.txt`).

No olvidar documentar apropiadamente los hechos y reglas en la base de datos.