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

Hechos, Reglas y Resolución

Este taller tiene por objetivos:

- Entender el proceso de resolución, utilizado por el Prolog para la solución de consultas.
- Entender y utilizar el operador *no es probable*, como mecanismo para resolver consultas negadas.
- Aprender a definir y utilizar reglas, las cuales posibilitan hacer programas haciendo uso de inferencias lógicas.

Ejemplo: Consideremos la siguiente base de datos:

```
mamifero(perro).
mamifero(gato).
mamifero(equidna).
mamifero(ornitorrinco).
oviparo(gallina).
oviparo(ornitorrinco).
oviparo(equidna).
```

Preguntas en forma afirmativa las puede resolver el Prolog,

```
?- mamifero(X).
X = perro ;
X = gato.

?- oviparo(Y).
Y = gallina ;
Y = ornitorrinco.
```

pero preguntas en forma negativa no:

```
?- \+mamifero(X).
false.

?- \+oviparo(Y).
false.
```

Esto ocurre porque el Prolog no puede particularizar variables libres, pues no tiene conocimiento del universo del discurso para los predicados que se le han definido.

Una solución parcial a este problema, para dominios finitos, es hacer explícito el universo del discurso en la base de datos. Podemos por ejemplo definir hechos de la siguiente forma:

```
animal(perro).
animal(gato).
animal(gallina).
animal(ornitorrinco).
animal(equidna).
```

De esta forma, podemos hacer una primera meta en forma positiva para particularizar la variable con un elemento del dominio y la segunda meta para la condición en forma negativa, por ejemplo:

```
?- animal(Y), \+oviparo(Y).
Y = perro ;
Y = gato ;
false.

?- animal(X), \+mamifero(X).
X = gallina ;
X = ornitorrinco.
```

Ejercicio 1

Retomar la base de datos de la práctica anterior con los hechos `picoyplaca/2`.

- Agregar los hechos necesarios para poder resolver consultas en forma negativa.
- Resolver la consulta: Qué dígitos no tienen pico y placa un determinado día?
- Resolver la consulta: Qué días no tiene pico y placa un determinado digito?

Reglas en Prolog

Una regla es la representación en Prolog de una implicación lógica, es decir un condicional que siempre es verdadero.

Para construir una regla se indica primero la conclusión, el símbolo :- y las condiciones para que la conclusión sea verdadera. Por ejemplo para decir que X y Y son hermanos si son hijos del mismo padre, lo indicamos:

```
hermanos(X,Y) :- padre(X,Z), padre(Y,Z), X\=Y.
```

Ejercicio 2

Extender la base de datos para incluir los siguientes hechos y reglas:

- Definir hechos `duennoDigito/2` que relacionen una persona con el último dígito de la placa de su carro.
- Definir la regla `distintoDia/2` que devuelve “true” si dos personas tienen pico y placa distinto dia de la semana.
- Definir la regla `carpool/4` que tomando como primer argumento el nombre de una persona, encuentra otra persona que no tenga picoyplaca el mismo dia y reporta los nombres de las dos personas y los dos días en que pueden hacer *carpooling*.

Ejercicio 3

Con base en los hechos y reglas definidos en el ejercicio 2 realizar las siguientes consultas:

- Qué dias tiene pico y placa una persona?
- Comprobar que `distintoDia` devuelve `true` para dos personas que no tienen pico y placa el mismo día.
- Comprobar que `distintoDia` devuelve `false` para dos personas que tienen pico y placa el mismo día.
- Indicar con quien puede hacer *carpooling* una persona y que dias de la semana.
- Ilustrar que ocurre cuando se busca una persona para hacer *carpooling*, pero no hay nadie más que tenga dias compatibles.
- Encontrar quienes no pueden hacer *carpooling* con una persona (porque le toca el pico y placa el mismo dia).

Informe

Nombrar la base de datos y las consultas siguiente el estándar <**NombreApellido>-<id>-BD4.pl** y <**NombreApellido>-<id>-consultas4.txt**. Remitir los 2 archivos al concluir la práctica.

No olvidar incluir los comentarios documentando el programa:

- Autor y fecha
- Descripción de cada uno de los predicados utilizados y sus argumentos.