Prolog
Ejecución y Depuración

Visión general

- La base de datos se almacena como un archivo (usualmente con extensión .pl)
- El entorno de Prolog “lee y compila” la base de datos cuando se le da el comando consultar

```
consult('programa').
```

```
['programa'].
```

- Si se hacen cambios a la base de datos, se puede volver a “consultar”.

Modo de consulta

- Una vez la base de datos se ha compilado, se pueden ejecutar consultas (queries) interactivamente
- Para salir del entorno se usa el predicado (sin argumentos):

```
halt.
```

Estructura de la base de datos

La base de datos se compone de:

- Hechos
- Reglas
- Comentarios: Se indican con el símbolo '%'

Desde el entorno de consultas se pueden ver los hechos y reglas actualmente en la base de datos:

```
listing.                % Muestra todo el contenido
listing(hermanos). % Muestra de definición de
                                la regla hermanos.
```

Ingreso de nuevas reglas a la BD

- Pasar al modo de ingreso de hechos y reglas a la BD:

```
[user].
```

- El prompt cambia por |: para indicar que esta aceptando entradas para la BD.
- Se digitan las nuevas reglas y se termina con la secuencia de “Fin de archivo” Ctrl-D

```
?- [user].
|: padre(x,y).
|: padre(z,y).
|: % user://3 compiled 0.00 sec, 824 bytes
true.
```

Seguimiento de la ejecución

- Ejecutar una consulta implicar encontrar una asignación de las variables y resolver todas sus cláusulas (unificación y resolución).
- Cuando se presentan errores resulta conveniente poder hacer un seguimiento del proceso de resolución y unificación a medida que el motor de Prolog intenta ejecutar la consulta. Esto se hace activando el modo de seguimiento:

```
trace.
```

- Se cancela el modo de seguimiento con

```
notrace.
```

Ejemplo de seguimiento

- Tenemos la siguiente base de datos:

```
% b,c son hijos de a
padre(b, a).
padre(c, a).

% e,f son hijos de d
padre(e, d).
padre(f, d).

% regla para definir hermanos
hermanos(X,Y) :- padre(X,Z), padre(Y,Z), X\=Y.
```

Seguimiento

```
?- ['padresHermanos'].
% padresHermanos compiled 0.00 sec, 0 bytes
true.

?- trace.
true.

[trace]  ?- hermanos(b,c).
   Call: (7) hermanos(b, c) ? creep
   Call: (8) padre(b, _G412) ? creep
   Exit: (8) padre(b, a) ? creep
   Call: (8) padre(c, a) ? creep
   Exit: (8) padre(c, a) ? creep
   Call: (8) b\=c ? creep
   Exit: (8) b\=c ? creep
   Exit: (7) hermanos(b, c) ? creep
true.
```

```
% regla para definir hermanos
hermanos(X,Y) :- padre(X,Z), padre(Y,Z), X\=Y.
```

Seguimiento

```
?- trace.
true.

[trace]  ?- hermanos(a,b).
   Call: (6) hermanos(a, b) ? creep
   Call: (7) padre(a, _G409) ? creep
   Fail: (7) padre(a, _G409) ? creep
   Fail: (6) hermanos(a, b) ? creep
false.
```

```
% regla para definir hermanos
hermanos(X,Y) :- padre(X,Z), padre(Y,Z), X\=Y.
```

Seguimiento

```
?- trace.
true.

[trace]  ?- hermanos(c,f).
   Call: (7) hermanos(c, f) ? creep
   Call: (8) padre(c, _G412) ? creep
   Exit: (8) padre(c, a) ? creep
   Call: (8) padre(f, a) ? creep
   Fail: (8) padre(f, a) ? creep
   Fail: (7) hermanos(c, f) ? creep
false.
```

```
% regla para definir hermanos
hermanos(X,Y) :- padre(X,Z), padre(Y,Z), X\=Y.
```