

8 La sección database

En la sección **database** se declaran los predicados que describen una base de datos interna. Una base de datos interna está compuesta de hechos que puedes añadir y borrar desde tu programa en tiempo de ejecución.

Los predicados que se declaran en la sección database se usan de la misma manera que los que se declaran en la sección predicates.

/* Programa 8.1 */

domains

nombre = symbol

edad = integer

database

persona(nombre,edad)

predicates

hombre(nombre,edad))

clauses

hombre(Nombre,Edad):-persona(Nombre,Edad).

En este ejemplo, puedes usar el predicado persona de la misma manera que usas el predicado hombre. La única diferencia es que puedes insertar y borrar hechos para persona mientras el programa se está ejecutando.

La sección database

Hay dos restricciones en el uso de los predicados declarados en la sección database.

1. Sólo se pueden añadir a la base de datos como hechos y no como reglas.
2. Los hechos en la base de datos no pueden tener variables libres.

Es posible tener diferentes secciones database, pero para hacer esto, debes de nombrar cada sección database.

database - mibasededatos

Esta declaración crea una base de datos con el nombre mibasededatos. Si no nombras a la sección se llamará por defecto **dbasedom**.

domains

nombre = symbol

edad = integer

database

persona(nombre,edad)

clauses

persona(elena,10).

persona(pedro,9).

Los hechos para predicados declarados en la sección database pueden ser especificados en tiempo de compilación en la sección clauses, como en le ejemplo anterior. En tiempo de ejecución los hechos pueden ser insertados y borrados, incluso los especificados en la sección clauses. No hay diferencias entre unos y otros.

En tiempo de ejecución, se pueden añadir hechos con los predicados **asserta**, **assertz** y **assert**, o grabando los hechos desde un fichero con el predicado **consult**.

1. ASSERTA. Añade un nuevo hecho en la base de datos, antes de los hechos que ya existen para el predicado dado.

asserta(hecho)

asserta(hecho,nombre de la base de datos)

2. ASSERTZ. Añade un nuevo hecho en la base de datos, después de los hechos que ya existen para el predicado dado.

assertz(hecho)

assertz(hecho,nombre de la base de datos)

3. CONSULT. Lee en un fichero hechos declarados en la sección database y los añade al final de la base de datos.

consult(nombre de fichero)

consult(nombre de fichero, nombre de la base de datos).

database

persona(symbol,edad)

predicates

datos

clauses

datos:-

```
assertz(persona(susana,35)),  
asserta(persona(juan,26)),  
assertz(persona(pepe,32)),  
asserta(persona(juan,26)).
```

También, podemos borrar hechos en tiempo de ejecución con los predicados **retract** y **retractall**

4. RETRACT Unifica un hecho y lo borra de la base de datos. En particular, borrará el primer hecho con el que se pueda unificar.

retract(hecho)

retract(hecho, basededatos)

5. RETRACTALL Borra todos los hechos de la base de datos que se puedan unificar.

retractall(hecho)

retractall(hecho,basededatos).

Con retractall no se pueden obtener valores de salida, esto significa que para variables no instanciadas debemos de usar la variable anónima.

```
retractall(persona(,-, -,masculino)).
```

Por último, podemos grabar los hechos de una base de datos con el predicado save.

6. SAVE.

```
save(nombre de fichero)
```

```
save(nombre de fichero, base de datos).
```