



[Predicado-havingObject] [Acciones condicionadas a los objetos]

A partir de la Fase 2 Plman puede realizar acciones condicionadas no sólo a lo que ve, sino a lo que lleva.

El predicado que controla si Plman lleva, o no, un objeto es "havingObject". Este predicado tiene dos modalidades, la que no tiene en cuenta el objeto que lleva y la que sí. El primero se declara como havingObject/0 (sin argumento), el segundo havingObject/1 (con un argumento que será el objeto).

Fases permitidas para usar havingObject >> fases 2, 3 y 4.

PREDICADO: havingObject/0

El predicado tiene **éxito** si Plman lleva un objeto (el que sea). **Fracasa** si no lo lleva.

Por ejemplo, si queremos que Plman deje a su izquierda el objeto que lleva podemos escribir:

```
do(drop(left)) :- havingObject, see(normal, left, ' ').
```

>> si Plman no lleva un objeto esta regla fracasa y no se ejecuta la acción.

PREDICADO: havingObject/1

El predicado tiene **éxito** si Plman lleva un objeto determinado, visible en su argumento. Se declara:

```
havingObject(appearance(Objeto)).
```

Objeto: objeto que puede llevar Plman

>> *el predicado tiene éxito si Plman lleva el objeto referido en el argumento "Objeto".*

Por ejemplo, si queremos que Plman use la llave "a" en la puerta "l", que se encuentra a su izquierda, podemos escribir:

```
do(use(left)) :- havingObject(appearance(a)), see(normal, left, '|').
```

>> *la regla se ejecutará si Plman lleva el objeto 'a' referido en el argumento.*

>> El predicado havingObject/n se escribe en el **cuerpo** de la regla. **NUNCA** en la cabeza. Evidente que sólo se puede incluir un predicado de este tipo pues Plman sólo puede llevar, como máximo, un objeto.

Por ejemplo, **no se pueden** declarar reglas con varios predicados havingObject, pq Plman sólo puede llevar un objeto:

```
do(use(left)) :- havingObject(appearance(a)), havingObject(appearance(r)), see(normal, left, '|'). (ERROR)
```