Práctica 1 – Programación declarativa

Autores:

Ignacio Amaya de la Peña – 11M021

Adrián Cámara Caunedo – 11M004

Borja Mas García – 11M001

En esta memoria vamos a incluir ejercicio por ejercicio los pasos que hemos seguido para resolver cada uno de ellos, así como las explicaciones del código utilizado para resolverlos.

PRIMERA PARTE

EJERCICIO 1

Para realizar este ejercicio basta con fijarnos en el dibujo proporcionado y utilizar los predicados que nos indican para así poder modelizar la figura. El código sería el siguiente:

```
en_suelo(a).

en_suelo(d).

en_suelo(f).

en_suelo(j).

contiguo(a,d).

contiguo(d,f).

contiguo(f,j).

sobre(c,d).

sobre(b,c).

sobre(e,f).

sobre(i,j).

sobre(h,i).

sobre(g,h).
```

EJERCICIO 2

Para este ejercicio debemos de realizar 3 predicados que nos sirvan para cualquier configuración y con ellos ver que elementos son base, cuales bases están a la derecha de otras y que objetos están a la derecha de otros. Para el predicado base(B1,B2) el código sería:

```
base(B1,B2):-
        B1=B2,
        en_suelo(B1).
base(B1,B2):-
        sobre(B1,X),
        base(X,B2).
```

Donde tenemos dos casos que serían el caso base, donde B1=B2 y que B1 este sobre el suelo, y el otro caso es en el que va yendo hacia abajo del elemento B1 y de forma recursiva llegamos al caso base y vemos el elemento de la base de la pila donde esta B1.

Para el predicado base_a_la_derecha(B1,B2) el código sería el siguiente:

```
base_a_la_derecha(B1,B2):-
        contiguo(B1,B2).
base_a_la_derecha(B1,B2):-
        contiguo(B1,X),
        base_a_la_derecha(X,B2).
```

Donde, como en el predicado anterior, tenemos dos casos. El primero sería el caso base donde el elemento B2 esta inmediatamente a la derecha del elemento B1. En el segundo caso iríamos cogiendo de forma recursiva el elemento contiguo de B1 y veríamos si éste es contiguo a B2 con el primer caso y sino volveríamos a ver el contiguo de éste y así sucesivamente para ver si el predicado del caso primero se hace cierto o no.

Por último estaría el predicado objeto_a_la_derecha(B1,B2), cuyo código sería:

```
objeto_a_la_derecha(B1,B2):-
        base(B1,X),
        base(B2,Y),
        base_a_la_derecha(X,Y).
```

Donde solo tenemos un caso, en el cual cogeríamos el elemento de la base donde esta B1 y el elemento de la base donde está B2, utilizando el predicado base descrito antes y con los elementos de la base utilizaríamos el predicado base_a_la_derecha para ver si la base de B2 está a la derecha de la base de B1 y, por tanto, B2 está a la derecha de B1.

EJERCICIO 3

3a) Con esta consulta tenemos ver todos los elementos que están a la izquierda de e, o lo que es lo mismo, todos los elementos que tienen a la derecha el objeto e. Para ello utilizamos en la consulta el predicado objeto_a_la_derecha(X,e), donde X tomará los valores de los elementos que tienen a la derecha el elemento e. La consulta sería la siguiente:

```
objeto_a_la_derecha(X,e).
X = a ;
X = d ;
X = c ;
X = b ;
```

Donde a, d, c, b son todos los objetos que están a la izquierda del elemento e.

3b) Para esta consulta necesitamos hacer una consulta anidada en la que preguntaremos por todos los elementos cuya base es la misma base que tiene el elemento b. Por tanto la consulta quedaría de la siguiente manera:

base(b,X),base(Y,X).

X = Y, Y = d ;

X = d,

Y = c ;

X = d,

Y = b ;

Donde los elementos que están en la misma base que b serían d, c y el propio b.

SEGUNDA PARTE

En esta segunda parte, trabajaremos con el mismo dibujo proporcionado, sólo que esta vez cada letra representará un objeto geométrico, como, por ejemplo, cubos, pirámides…

EJERCICIO 4

En este ejercicio, el primero de la segunda parte de la práctica, lo que debemos realizar es un predicado que asigne a cada letra un objeto determinado. En nuestro caso, este predicado será de la forma "forma(X,Y)" asignando a cada X una letra de la 'a' a la 'j' y asignando a cada Y un objeto que se le asociará a su letra respectiva. Así pues, el código quedaría de la siguiente forma:

forma(a,piramide).

forma(b,toro).

forma(c,cubo).

forma(d,esfera).

forma(e,cubo).

forma(f,toro).

forma(g,toro).

forma(h,esfera).

forma(i,toro).

forma(j,piramide).

EJERCICIO 5

El siguiente ejercicio consiste en, dadas varias formas apiladas de distintas maneras, crear un predicado que examine si es posible o no apilarlos de ese modo. En otras palabras, ver si es inestable. Para programar el predicado son necesarios cinco casos distintos que iremos explicando a la par que exponemos su código correspondiente.

- Caso 1:

```
inestable(O):-
        forma(O,X),
        X\=toro,
        forma(Y,piramide),
        sobre(O,Y).
```

En este primer caso analizamos la inestabilidad de colocar objetos encima de una pirámide. Como bien nos dice el enunciado, sobre la pirámide sólo puede haber un toro, y cualquier otro objeto sería inestable sobre ella. Por tanto, en primer lugar lo que hacemos es comprobar la forma de nuestro objeto. Si nuestro objeto es distinto de un toro, y además al hacer la comprobación de que el objeto está sobre la pirámide es verdadera, entonces nuestro predicado nos dirá que esa configuración es inestable, y en caso de ser un toro, dirá que es estable.

- Caso 2:

```
inestable(O):-
        forma(O,X),
        X\=toro,
        forma(Y,piramide),
        forma(Z,toro),
        sobre(Z,Y),
        sobre(O,Z).
```

Este caso es una continuación directa del caso 1, puesto que en una pirámide, antes de colocar cualquier otro objeto, debemos colocar antes 2 toros(Ya que sólo con 1 la punta de la pirámide sobresaldría y generaría una inestabilidad con cualquier otro objeto). Por tanto, en este caso hacemos algo similar al anterior, verificar cuál es el segundo objeto que colocamos en la pirámide. Por tanto, lo que hacemos con este código es lo siguiente: Verificamos cuál es nuestro objeto, y, si no es un toro y la comprobación de que los dos objetos anteriores son un toro y, debajo de éste una pirámide, es verdadera, entonces este predicado dirá que es inestable, y viceversa en caso de que el nuestro objeto sea un toro.

- Caso 3:

```
inestable(O):-
        forma(O,esfera),
        en_suelo(O).
```

Este caso es muy sencillo, ya que lo único que debemos comprobar es si nuestro objeto es una esfera y si ésta está en el suelo. De ser así, entonces diremos que nuestra configuración es inestable.

- Caso 4:

```
inestable(O):-
        forma(O,esfera),
        sobre(O,X),
        forma(X,Y),
        Y\=toro.
```

En este caso, continuando con la inestabilidad de la esfera, comprobaremos si nuestra esfera está sobre un toro o sobre cualquier otro objeto. En caso de estar sobre un toro, entonces será estable, y en caso contrario, inestable. Para ello y como bien se puede apreciar en el código, comprobamos en primer lugar que nuestro objeto es una esfera. Tras ello, comprobamos la forma sobre la que está nuestra esfera, y, en caso de no ser un toro, nuestro predicado nos devolverá que es inestable.

- Caso 5:

```
inestable(O):-
        forma(O,X),
        X\=toro,
        sobre(O,Y),
        forma(Y,esfera).
```

En este último caso, y continuando con la inestabilidad que genera la esfera, comprobaremos que encima de ésta sólo tiene cabida colocar un toro. Para ello, comprobamos en primer lugar que objeto es el que tenemos. En caso de no ser un toro, comprobamos que nuestro objeto está sobre otro objeto, y, además, éste objeto es una esfera. De ser así, entonces nuestro predicado nos devolverá inestable, y en caso de que nuestro objeto sea un toro, nos devolverá que es estable.

EJERCICIO 6

En este ejercicio debemos comprobar la estabilidad de la configuración inicial propuesta utilizando nuestro predicado programado en el ejercicio 5. Así pues, al comprobar la estabilidad resultó que la configuración tenía 2 puntos de inestabilidad claros. Los objetos d,c, y b, que son respectivamente una esfera, un cubo y un toro, estaban colocados de manera errónea. La colocación inicial era una esfera en el suelo, sobre la esfera colocaba un cubo y finalmente, en la cima un toro. Por tanto generaba inestabilidad el hecho de tener una esfera

sobre el suelo y también generaba inestabilidad tener un cubo sobre una esfera, ya que sobre la esfera sólo podemos colocar un toro. Por tanto para corregir esto lo único que debíamos hacer era colocar el cubo en el suelo, sobre éste el toro y en la cima la esfera, eliminando así la inestabilidad.

Por otra parte, en el bloque compuesto por g, h, i, j que son respectivamente un toro, una esfera, otro toro y una pirámide, generaba una clara inestabilidad. Esta inestabilidad estaba generada porque, al tener una pirámide en la base, después de ella debemos colocar 2 toros, y en este caso tenemos un toro y una esfera. La solución era simple, permutar la esfera y el toro superior, y colocarlos de manera que quedara una pirámide, luego dos toros y finalmente, en la parte superior, una esfera.

Así pues, la configuración estable generada será de la forma:

en_suelo(a).

en_suelo(c).

en_suelo(f).

en_suelo(j).

contiguo(a,c).

contiguo(c,f).

contiguo(f,j).

sobre(b,c).

sobre(d,b).

sobre(e,f).

sobre(i,j).

sobre(g,i).

sobre(h,g).