

Técnicas de los Sistemas Inteligentes

Práctica 3 – Entrega 1

Curso 2016/2017

Francisco Javier Caracuel Beltrán

**Grupo 3
Grado en Ingeniería Informática - CCIA**

Índice

1. Ejercicio 1: 3

2. Ejercicio 2:8

3. Ejercicio 3: 11

4. Ejercicio 4: 17

5. Ejercicio 5: 21

1. Ejercicio 1:

- a) Para representar los objetos del mundo se ha decidido crear 3 tipos principales: *element*, *zone* y *orientation*.
Element a su vez se divide en *person* y *obj* y, *person* se divide en *player* y *character*.
 Las zonas del mundo serán *zone*, *orientation* será la orientación de las zonas y del jugador y, *obj* son los objetos situados sobre las zonas del mapa, *player* es el jugador y *character* es cada personaje.
 Se ha seguido esta jerarquía porque permite expresar el funcionamiento del mundo con el uso de menos predicados.
- b) Para describir los estados del mundo se utilizan los siguientes predicados:
- *orientated zone1 orientation zone2*
 Indica que la zona 1 está orientada al norte/sur/este/oeste sobre la zona 2.
 No se describe que una zona está conectada con otra, porque se entiende que, si una zona está orientada de alguna manera sobre otra, es cuando están conectadas.
 - *lookAt player orientation*
 Indica que el jugador está mirando hacia el norte/sur/este/oeste.
 - *at element zone*
 Indica que el jugador/personaje/objeto están colocados sobre una zona en concreto.
 - *has person obj*
 Indica que el jugador/personaje tiene un objeto.
 - *EH*
 Para controlar que el jugador tiene la mano vacía o no. Al principio puede que no sea imprescindible, pero será de utilidad controlarlo.
- c) Las acciones son:
- *TURN_RIGHT/TURN_LEFT*:
 Las dos acciones son iguales, lo único que cambia es hacia dónde mira el jugador.
 No tiene precondiciones porque no le afecta en nada lo que haya sobre el mapa para girar.
 En el efecto se utiliza un esquema condicional por el que se indica que, si mira hacia un lado, al girar cambiará de lado al que mira y, además, hay que indicar que ya no está mirando hacia el lado inicial.
 - *GO*:
 Las precondiciones son que la orientación del jugador debe ser la misma que la orientación que tiene la zona a la que quiere ir sobre la zona en la que está.
 Los efectos son que el jugador está en la zona nueva y ya no está en la zona antigua.
 - *GET*:
 Para coger un objeto debe tener la mano vacía, y el objeto encontrarse en la misma zona en la que se encuentra el jugador.
 Los efectos son que el jugador no tiene la mano vacía, el objeto ya no se encuentra en esa zona y que el jugador tiene el objeto.

- **RELEASE:**

Para soltar un objeto, el jugador debe tenerlo, no puede tener la mano vacía y se debe indicar que el jugador esté sobre una zona, para que así en el efecto el objeto se deje sobre esa misma zona. Si no se pone esta última precondition, el objeto se soltaría en otra distinta.

Los efectos son que el jugador no tiene el objeto, la mano está vacía y el objeto está sobre la misma zona en la que está el jugador.

- **GIVE:**

Para dar un objeto a un personaje, el jugador debe tener el objeto, la mano no puede estar vacía y el jugador y el personaje deben estar en la misma zona.

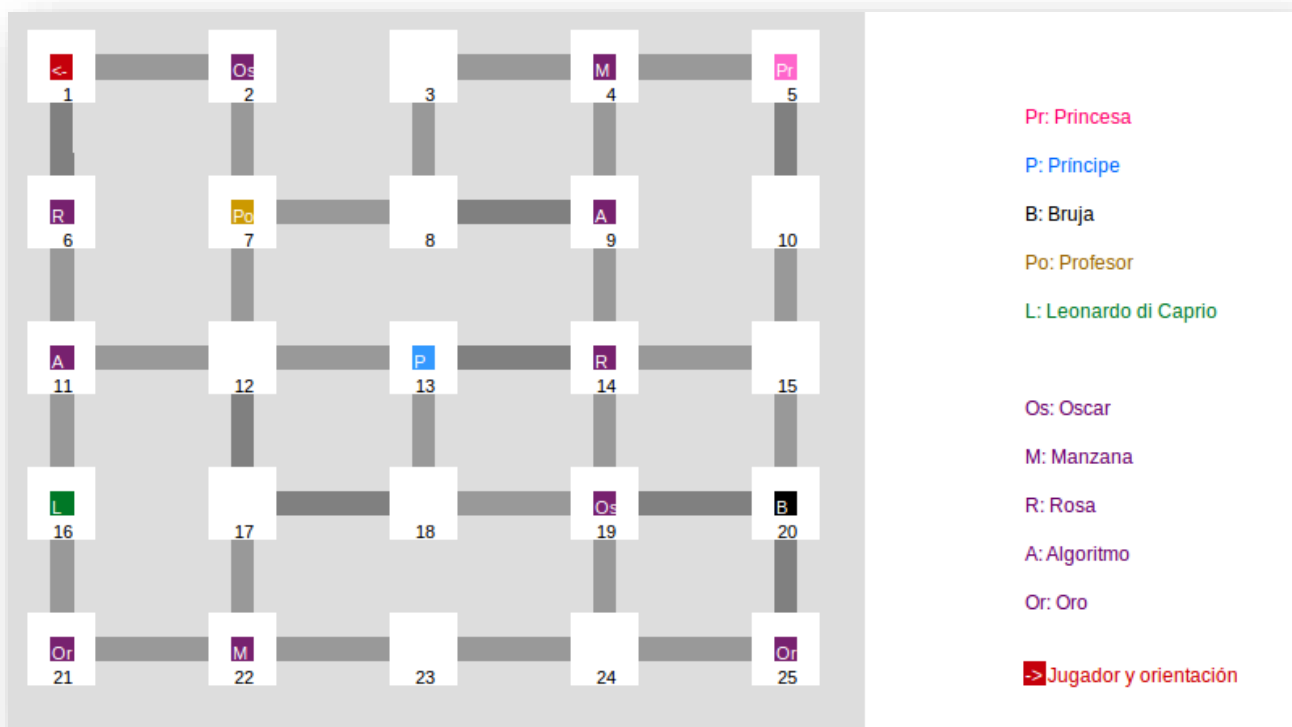
Los efectos son que el jugador no tiene el objeto, tiene la mano vacía y el personaje tiene el objeto.

- d) En el problema se indica que existe un jugador, los cinco personajes, los cinco objetos, las cuatro orientaciones y las veinticinco zonas.

Como, por defecto, se tiene en cuenta que nada es verdad, como inicio hay que indicar que el jugador está en una zona, su orientación y que tiene la mano vacía. También se indican las posiciones de todos los personajes y objetos y se indica que orientación tiene cada zona con la adyacente, de esta forma se define como es la topología del mapa.

El objetivo es que todos los objetos deben tenerlos algún personaje. Para indicar esto, se hace uso de *exists*, en el que se define cinco objetos y se establece que cada personaje debe tener uno de ellos.

El mapa que se ha creado para desarrollar este ejercicio es el siguiente:



La versión original del mapa tenía muchas menos conexiones, pero cuando se ha intentado ejecutar la planificación con optimización el tiempo de cómputo era muy grande.

Para solucionar este hecho, se han añadido más conexiones y el doble de objetos, con el fin de facilitar un poco la ejecución.

El mapa final y utilizado ha sido el que se muestra.

El resultado de ejecutar la planificación es (con optimización, -g=1, -h=1):

```
ff: found legal plan as follows
step    0: TURN_LEFT P
        1: TURN_LEFT P
        2: GO P Z1 Z2 EAST
        3: TURN_RIGHT P
        4: GET P OSCAR Z2
        5: GO P Z2 Z7 SOUTH
        6: GIVE P PROFESSOR OSCAR Z7
        7: GO P Z7 Z12 SOUTH
        8: GO P Z12 Z17 SOUTH
        9: GO P Z17 Z22 SOUTH
       10: TURN_RIGHT P
       11: GO P Z22 Z21 WEST
       12: TURN_RIGHT P
       13: GET P GOLD Z21
       14: GO P Z21 Z16 NORTH
       15: GIVE P LEO GOLD Z16
       16: GO P Z16 Z11 NORTH
       17: GET P ALGORITHM Z11
       18: TURN_RIGHT P
       19: GO P Z11 Z12 EAST
       20: GO P Z12 Z13 EAST
       21: GIVE P PRINCE ALGORITHM Z13
       22: GO P Z13 Z14 EAST
       23: GET P ROSE Z14
       24: GO P Z14 Z15 EAST
       25: TURN_RIGHT P
       26: GO P Z15 Z20 SOUTH
       27: GIVE P WITCH ROSE Z20
       28: GO P Z20 Z25 SOUTH
       29: GET P GOLD Z25
       30: TURN_RIGHT P
       31: TURN_RIGHT P
       32: GO P Z25 Z20 NORTH
       33: GO P Z20 Z15 NORTH
       34: GO P Z15 Z10 NORTH
       35: GO P Z10 Z5 NORTH
       36: GIVE P PRINCESS GOLD Z5
       37: REACH-GOAL
           Coste Total: 0.00

time spent:    0.16 seconds instantiating 945 easy, 0 hard action templates
               0.00 seconds reachability analysis, yielding 191 facts and 3470 actions
               0.06 seconds creating final representation with 187 relevant facts, 0 relevant fluents
               0.00 seconds computing LNF
               0.00 seconds building connectivity graph
              17.30 seconds searching, evaluating 31645 states, to a max depth of 0
              17.52 seconds total time
```

Debido a la falta de tiempo de la que se dispone, no es posible crear un nuevo mapa con todas sus conexiones, por lo que para hacer pruebas se ha optado por cambiar el objetivo o la ubicación de algunos objetos y/o personajes.

Una muestra de la ejecución cuyo objetivo es dar a cada personaje el objeto que le corresponde es (sin optimización):

```
ff: found legal plan as follows
step  0: TURN_LEFT P
      1: GO P Z1 Z6 SOUTH
      2: GO P Z6 Z11 SOUTH
      3: GO P Z11 Z16 SOUTH
      4: GO P Z16 Z21 SOUTH
      5: TURN_LEFT P
      6: GO P Z21 Z22 EAST
      7: GO P Z22 Z23 EAST
      8: GO P Z23 Z24 EAST
      9: GO P Z24 Z25 EAST
     10: TURN_LEFT P
     11: GO P Z25 Z20 NORTH
     12: GO P Z20 Z15 NORTH
     13: TURN_LEFT P
     14: GO P Z15 Z14 WEST
     15: TURN_RIGHT P
     16: TURN_RIGHT P
     17: GET P ROSE Z14
     18: GO P Z14 Z15 EAST
     19: TURN_LEFT P
     20: GO P Z15 Z10 NORTH
     21: GO P Z10 Z5 NORTH
     22: TURN_LEFT P
     23: GIVE P PRINCESS ROSE Z5
     24: GO P Z5 Z4 WEST
     25: TURN_LEFT P
     26: GET P APPLE Z4
     27: GO P Z4 Z9 SOUTH
     28: GO P Z9 Z14 SOUTH
     29: GO P Z14 Z19 SOUTH
     30: TURN_LEFT P
     31: GO P Z19 Z20 EAST
     32: TURN_LEFT P
     33: GIVE P WITCH APPLE Z20
     34: GO P Z20 Z15 NORTH
     35: TURN_LEFT P
     36: GO P Z15 Z14 WEST
     37: GO P Z14 Z13 WEST
     38: GO P Z13 Z12 WEST
     39: GO P Z12 Z11 WEST
     40: TURN_LEFT P
     41: GET P ALGORITHM Z11
     42: GO P Z11 Z16 SOUTH
     43: GO P Z16 Z21 SOUTH
     44: TURN_LEFT P
     45: TURN_LEFT P
     46: RELEASE P ALGORITHM Z21
     47: GET P GOLD Z21
     48: GO P Z21 Z16 NORTH
     49: GO P Z16 Z11 NORTH
     50: TURN_RIGHT P
```

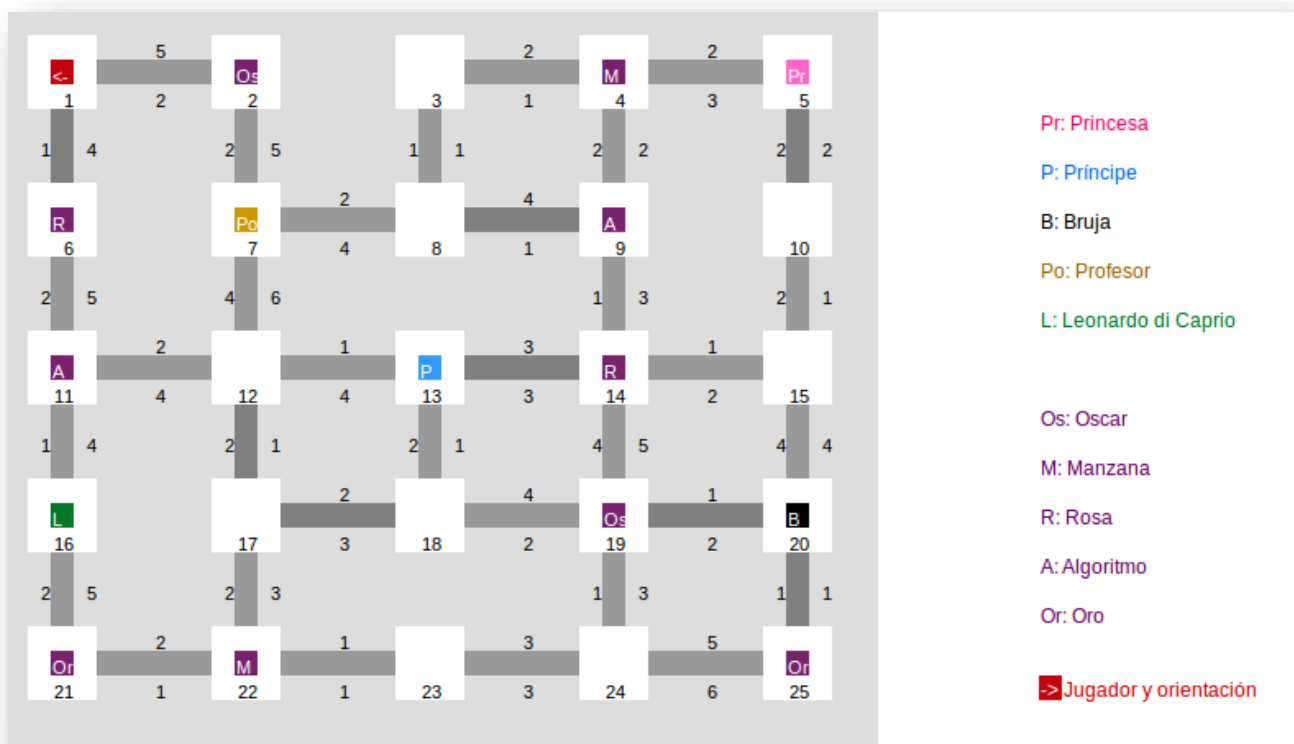
```
50: TURN_RIGHT P
51: GO P Z11 Z12 EAST
52: GO P Z12 Z13 EAST
53: GIVE P PRINCE GOLD Z13
54: GO P Z13 Z14 EAST
55: TURN_LEFT P
56: GO P Z14 Z9 NORTH
57: GET P ALGORITHM Z9
58: TURN_LEFT P
59: GO P Z9 Z8 WEST
60: GO P Z8 Z7 WEST
61: GIVE P PROFESSOR ALGORITHM Z7
62: TURN_LEFT P
63: GO P Z7 Z12 SOUTH
64: TURN_LEFT P
65: GO P Z12 Z13 EAST
66: TURN_RIGHT P
67: GO P Z13 Z18 SOUTH
68: TURN_LEFT P
69: GO P Z18 Z19 EAST
70: GET P OSCAR Z19
71: GO P Z19 Z20 EAST
72: TURN_LEFT P
73: GO P Z20 Z15 NORTH
74: TURN_LEFT P
75: GO P Z15 Z14 WEST
76: GO P Z14 Z13 WEST
77: GO P Z13 Z12 WEST
78: GO P Z12 Z11 WEST
79: TURN_LEFT P
80: GO P Z11 Z16 SOUTH
81: GIVE P LEO OSCAR Z16
82: GO P Z16 Z21 SOUTH
83: TURN_LEFT P
84: GO P Z21 Z22 EAST
85: GO P Z22 Z23 EAST
86: GO P Z23 Z24 EAST
87: GO P Z24 Z25 EAST
88: TURN_LEFT P
89: GO P Z25 Z20 NORTH
    Coste Total: 0.00
```

```
time spent:    0.00 seconds instantiating 945 easy, 0 hard action templates
              0.00 seconds reachability analysis, yielding 191 facts and 345 actions
              0.00 seconds creating final representation with 186 relevant facts, 0 relevant fluents
              0.00 seconds computing LNF
              0.00 seconds building connectivity graph
              0.11 seconds searching, evaluating 2853 states, to a max depth of 9
              0.11 seconds total time
```

2. Ejercicio 2:

- Para adecuar el dominio a la nueva restricción de desplazamiento, se ha creado la función *distance zone zone*, que indica el coste de ir de una zona a otra del mapa.
- En el problema se han establecido nuevas relaciones iniciales, indicando la distancia existente entre zonas del mapa. Se ha indicado una distancia diferente dependiendo de la dirección que tome el jugador, es decir, tiene un coste diferente ir de la zona 1 a la zona 2, que ir de la zona 2 a la zona 1.
- Para encontrar el camino mínimo, se crea una función que tendrá el valor total que ha tomado el jugador en cada ruta.
El objetivo de esta función es poder usarla con *metric*, para indicarle que debe minimizarla. Se debe añadir a la acción *GO* un efecto por el que cuando avance de zona, se incremente usando la función creada en el apartado A.

El mapa modificado para albergar la distancia entre zonas es el siguiente:



El resultado de ejecutar la planificación es (con optimización, -g=1, -h=1):

```
ff: found legal plan as follows
step    0: TURN_LEFT P
        1: TURN_LEFT P
        2: GO P Z1 Z2 EAST
        3: TURN_RIGHT P
        4: GET P OSCAR Z2
        5: GO P Z2 Z7 SOUTH
        6: TURN_LEFT P
        7: GIVE P PROFESSOR OSCAR Z7
        8: GO P Z7 Z8 EAST
        9: TURN_LEFT P
       10: GO P Z8 Z3 NORTH
       11: TURN_RIGHT P
       12: GO P Z3 Z4 EAST
       13: GET P APPLE Z4
       14: GO P Z4 Z5 EAST
       15: RELEASE P APPLE Z5
       16: TURN_RIGHT P
       17: TURN_RIGHT P
       18: GET P APPLE Z5
       19: GIVE P PRINCESS APPLE Z5
       20: GO P Z5 Z4 WEST
       21: TURN_LEFT P
       22: GO P Z4 Z9 SOUTH
       23: GET P ALGORITHM Z9
       24: GO P Z9 Z14 SOUTH
       25: TURN_RIGHT P
       26: GO P Z14 Z13 WEST
       27: GIVE P PRINCE ALGORITHM Z13
       28: TURN_LEFT P
       29: GO P Z13 Z18 SOUTH
       30: TURN_LEFT P
       31: GO P Z18 Z19 EAST
       32: GET P OSCAR Z19
       33: GO P Z19 Z20 EAST
       34: TURN_LEFT P
       35: GIVE P WITCH OSCAR Z20
       36: TURN_LEFT P
       37: TURN_LEFT P
       38: GO P Z20 Z25 SOUTH
       39: GET P GOLD Z25
       40: TURN_RIGHT P
       41: TURN_RIGHT P
       42: GO P Z25 Z20 NORTH
       43: GO P Z20 Z15 NORTH
       44: TURN_LEFT P
       45: GO P Z15 Z14 WEST
       46: GO P Z14 Z13 WEST
       47: GO P Z13 Z12 WEST
       48: GO P Z12 Z11 WEST
       49: RELEASE P GOLD Z11
       50: GET P ALGORITHM Z11
```

```

51: TURN_LEFT P
52: GO P Z11 Z16 SOUTH
53: RELEASE P ALGORITHM Z16
54: TURN_RIGHT P
55: TURN_RIGHT P
56: GET P ALGORITHM Z16
57: GIVE P LEO ALGORITHM Z16
58: REACH-GOAL
    Coste Total: 41.00

time spent:    0.17 seconds instantiating 945 easy, 0 hard action templates
               0.00 seconds reachability analysis, yielding 191 facts and 3470 actions
               0.06 seconds creating final representation with 187 relevant facts, 1 relevant fluents
               0.01 seconds computing LNF
               0.00 seconds building connectivity graph
               68.58 seconds searching, evaluating 94018 states, to a max depth of 0
               68.82 seconds total time

```

Para hacer experimentos y comprobar que funciona, lo que se ha cambiado ha sido la distancia entre la zona 1 y la zona 2, estableciendo un valor de 2.000. De este modo, el primer avance que era ir de la zona 1 a la zona 2, ya no ocurre y va hacia la zona 6:

```

ff: found legal plan as follows

step    0: TURN_LEFT P
        1: GO P Z1 Z6 SOUTH
        2: GO P Z6 Z11 SOUTH
        3: GET P ALGORITHM Z11
        4: GO P Z11 Z16 SOUTH
        5: GIVE P LEO ALGORITHM Z16
        6: GO P Z16 Z21 SOUTH
        7: GET P GOLD Z21
        8: TURN_LEFT P
        9: GO P Z21 Z22 EAST
       10: TURN_LEFT P
       11: GO P Z22 Z17 NORTH
       12: GO P Z17 Z12 NORTH
       13: GO P Z12 Z7 NORTH
       14: TURN_RIGHT P
       15: GIVE P PROFESSOR GOLD Z7
       16: GO P Z7 Z8 EAST
       17: GO P Z8 Z9 EAST
       18: GET P ALGORITHM Z9
       19: TURN_LEFT P
       20: GO P Z9 Z4 NORTH
       21: TURN_LEFT P
       22: TURN_LEFT P
       23: RELEASE P ALGORITHM Z4
       24: GET P APPLE Z4
       25: TURN_LEFT P
       26: GO P Z4 Z5 EAST
       27: GIVE P PRINCESS APPLE Z5
       28: TURN_RIGHT P
       29: GO P Z5 Z10 SOUTH
       30: GO P Z10 Z15 SOUTH
       31: TURN_RIGHT P
       32: GO P Z15 Z14 WEST
       33: GET P ROSE Z14
       34: GO P Z14 Z13 WEST
       35: TURN_LEFT P
       36: GIVE P PRINCE ROSE Z13
       37: GO P Z13 Z18 SOUTH
       38: TURN_LEFT P
       39: GO P Z18 Z19 EAST
       40: GET P OSCAR Z19
       41: GO P Z19 Z20 EAST
       42: RELEASE P OSCAR Z20
       43: TURN_RIGHT P
       44: TURN_RIGHT P
       45: GET P OSCAR Z20
       46: GIVE P WITCH OSCAR Z20
       47: REACH-GOAL
    Coste Total: 41.00

time spent:    0.17 seconds instantiating 945 easy, 0 hard action templates
               0.00 seconds reachability analysis, yielding 191 facts and 3470 actions
               0.05 seconds creating final representation with 187 relevant facts, 1 relevant fluents
               0.00 seconds computing LNF
               0.01 seconds building connectivity graph
               3.08 seconds searching, evaluating 8394 states, to a max depth of 0
               3.31 seconds total time

```

3. Ejercicio 3:

- a) Se ha creado un nuevo objeto llamado *tool*, que hereda de *obj*. Las zapatillas y el bikini serán de tipo *tool*.

Para indicar que una zona es de un tipo dado, se crea el predicado *is zona tipo*.

El jugador debe poder acceder a una zona solo si tiene las zapatillas en el bosque o el bikini en el agua, por lo que en la acción *GO* se añade una precondition más que diga que puede pasar si la zona no es ni bosque ni agua, o si la zona es bosque tiene que tener las *zapatillas*, o si la zona es agua tiene que tener el *bikini*, o si es precipicio no puede pasar.

Los casos anteriores se establecen con la cláusula *or* y tendrá tantas condiciones como se ha descrito anteriormente.

- b) Para poder meter y sacar objetos de la mochila se ha creado el predicado *backpack obj*. Como solo se puede tener un objeto en la mano y un objeto en la mochila a la vez, se ha tenido que modificar la acción *RELEASE*, indicando que no se puede dejar un objeto que esté en la mochila. También se ha modificado la acción *GET*, para establecer que no se puede coger un objeto que ya esté en la mochila (en la acción solo se pone *has player obj*, ya que, si tiene la mano vacía y tiene un objeto, implícitamente lo tiene en la mochila).

Se crean dos nuevas acciones:

- *PUT_BACKPACK*:

Para poner en la mochila un objeto, la mano no puede estar vacía, tiene que tener el objeto y no puede haber un objeto cualquiera en la mochila.

El efecto es que la mano está vacía y en la mochila está el objeto.

- *POP_BACKPACK*:

Para sacar un objeto de la mochila, el jugador no puede tener ningún objeto en la mano y tiene que tener en la mochila el objeto.

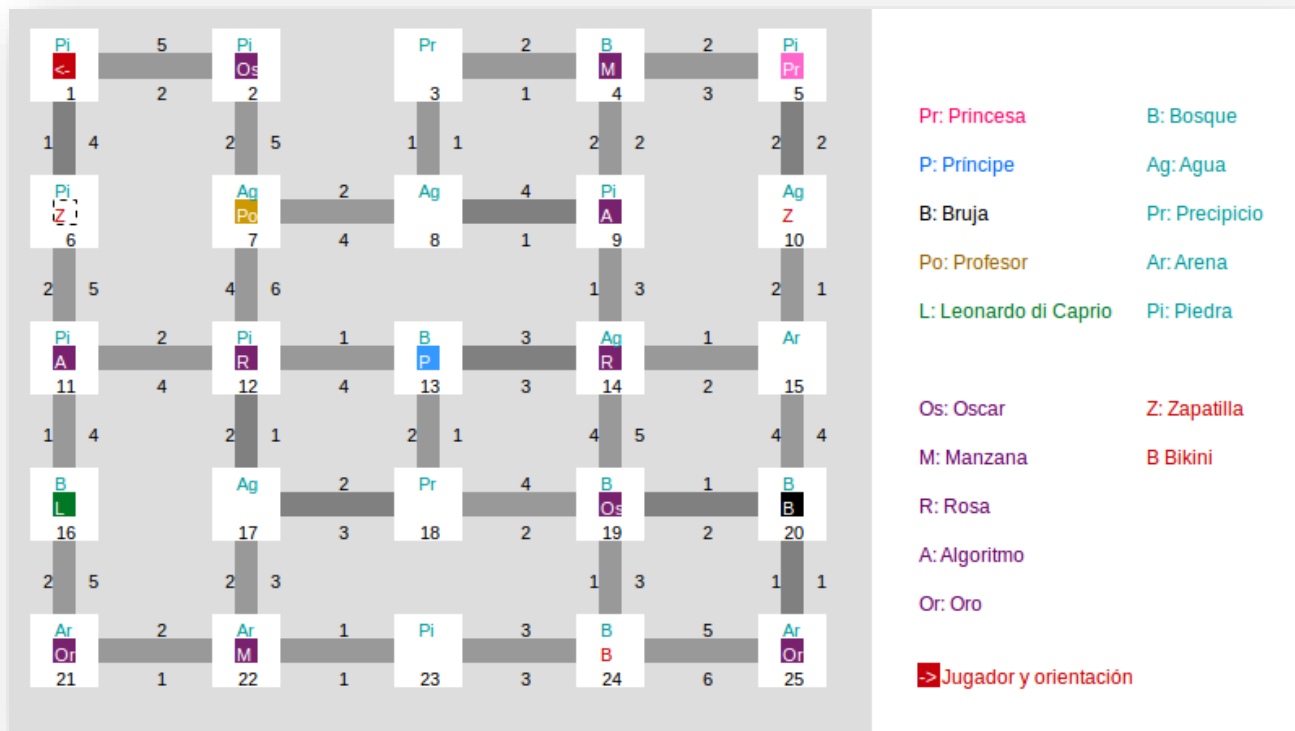
Los efectos son que no tiene el objeto en la mochila y la mano no está vacía.

- c) En el problema se han declarado las zapatillas y el bikini como tipo *tool*.

Como inicialización se ha indicado para cada zona, el tipo de terreno que es.

Para poder ir por todo el mapa, también se añade en qué posición están las zapatillas y el bikini.

Con las nuevas modificaciones, el mapa queda como sigue:



Se ha añadido la restricción del apartado c y se establecen unas zapatillas en la zona 10 donde hay agua y un bikini en la zona 24 donde hay bosque.

El resultado de ejecutar la planificación es (con optimización, -g=1, -h=1):

```
ff: found legal plan as follows

step    0: TURN_LEFT P
        1: GO P Z1 Z6 SOUTH
        2: TURN_RIGHT P
        3: TURN_RIGHT P
        4: GET P SHOE Z6
        5: PUT_BACKPACK P SHOE
        6: TURN_LEFT P
        7: TURN_LEFT P
        8: GO P Z6 Z11 SOUTH
        9: GET P ALGORITHM Z11
       10: GO P Z11 Z16 SOUTH
       11: GIVE P LEO ALGORITHM Z16
       12: GO P Z16 Z21 SOUTH
       13: TURN_RIGHT P
       14: TURN_RIGHT P
       15: POP_BACKPACK P SHOE
       16: RELEASE P SHOE Z21
       17: GET P GOLD Z21
       18: PUT_BACKPACK P GOLD
       19: GET P SHOE Z21
       20: TURN_RIGHT P
       21: GO P Z21 Z22 EAST
       22: GO P Z22 Z23 EAST
       23: RELEASE P SHOE Z23
       24: POP_BACKPACK P GOLD
       25: RELEASE P GOLD Z23
       26: GET P SHOE Z23
       27: PUT_BACKPACK P SHOE
       28: GET P GOLD Z23
       29: GO P Z23 Z24 EAST
       30: RELEASE P GOLD Z24
       31: TURN_LEFT P
       32: GET P BIKINI Z24
       33: GO P Z24 Z19 NORTH
       34: RELEASE P BIKINI Z19
       35: POP_BACKPACK P SHOE
       36: RELEASE P SHOE Z19
       37: GET P BIKINI Z19
       38: PUT_BACKPACK P BIKINI
       39: GET P SHOE Z19
       40: TURN_RIGHT P
       41: GO P Z19 Z20 EAST
       42: RELEASE P SHOE Z20
       43: POP_BACKPACK P BIKINI
       44: RELEASE P BIKINI Z20
       45: GET P SHOE Z20
       46: PUT_BACKPACK P SHOE
       47: GET P BIKINI Z20
       48: TURN_LEFT P
       49: GIVE P WITCH SHOE Z20
       50: GO P Z20 Z15 NORTH
```



```
51: GO P Z15 Z10 NORTH
52: GET P SHOE Z10
53: GO P Z10 Z5 NORTH
54: RELEASE P BIKINI Z5
55: TURN_LEFT P
56: GO P Z5 Z4 WEST
57: GET P APPLE Z4
58: TURN_RIGHT P
59: TURN_RIGHT P
60: RELEASE P APPLE Z4
61: POP_BACKPACK P SHOE
62: RELEASE P SHOE Z4
63: GET P APPLE Z4
64: PUT_BACKPACK P APPLE
65: GET P SHOE Z4
66: GO P Z4 Z5 EAST
67: TURN_RIGHT P
68: RELEASE P SHOE Z5
69: POP_BACKPACK P APPLE
70: RELEASE P APPLE Z5
71: GET P SHOE Z5
72: PUT_BACKPACK P SHOE
73: GET P APPLE Z5
74: GIVE P PRINCESS APPLE Z5
75: POP_BACKPACK P SHOE
76: TURN_RIGHT P
77: TURN_RIGHT P
78: PUT_BACKPACK P SHOE
79: GET P BIKINI Z5
80: TURN_LEFT P
81: GO P Z5 Z4 WEST
82: RELEASE P BIKINI Z4
83: POP_BACKPACK P SHOE
84: RELEASE P SHOE Z4
85: GET P BIKINI Z4
86: PUT_BACKPACK P BIKINI
87: GET P SHOE Z4
88: TURN_LEFT P
89: GO P Z4 Z9 SOUTH
90: RELEASE P SHOE Z9
91: POP_BACKPACK P BIKINI
92: RELEASE P BIKINI Z9
93: GET P ALGORITHM Z9
94: PUT_BACKPACK P ALGORITHM
95: GET P SHOE Z9
96: TURN_RIGHT P
97: TURN_RIGHT P
98: RELEASE P SHOE Z9
99: POP_BACKPACK P ALGORITHM
100: RELEASE P ALGORITHM Z9
```

```

101: GET P SHOE Z9
102: PUT_BACKPACK P SHOE
103: GET P ALGORITHM Z9
104: TURN_RIGHT P
105: TURN_RIGHT P
106: RELEASE P ALGORITHM Z9
107: GET P BIKINI Z9
108: GO P Z9 Z14 SOUTH
109: TURN_RIGHT P
110: GO P Z14 Z13 WEST
111: RELEASE P BIKINI Z13
112: POP_BACKPACK P SHOE
113: RELEASE P SHOE Z13
114: GET P BIKINI Z13
115: PUT_BACKPACK P BIKINI
116: GET P SHOE Z13
117: GIVE P PRINCE SHOE Z13
118: POP_BACKPACK P BIKINI
119: GO P Z13 Z12 WEST
120: RELEASE P BIKINI Z12
121: GET P ROSE Z12
122: PUT_BACKPACK P ROSE
123: GET P BIKINI Z12
124: TURN_RIGHT P
125: TURN_RIGHT P
126: RELEASE P BIKINI Z12
127: POP_BACKPACK P ROSE
128: RELEASE P ROSE Z12
129: GET P BIKINI Z12
130: PUT_BACKPACK P BIKINI
131: GET P ROSE Z12
132: TURN_LEFT P
133: GO P Z12 Z7 NORTH
134: RELEASE P ROSE Z7
135: POP_BACKPACK P BIKINI
136: RELEASE P BIKINI Z7
137: GET P ROSE Z7
138: PUT_BACKPACK P ROSE
139: TURN_RIGHT P
140: TURN_RIGHT P
141: POP_BACKPACK P ROSE
142: RELEASE P ROSE Z7
143: GET P BIKINI Z7
144: PUT_BACKPACK P BIKINI
145: GET P ROSE Z7
146: GIVE P PROFESSOR ROSE Z7
147: REACH-GOAL
      Coste Total: 43.00

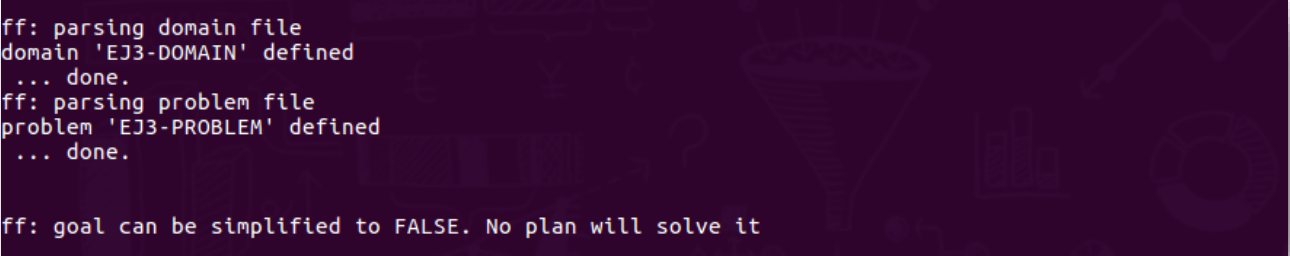
```

```

time spent:  6.12 seconds instantiating 1241 easy, 63 hard action templates
            0.00 seconds reachability analysis, yielding 293 facts and 17238 actions
            3.05 seconds creating final representation with 289 relevant facts, 1 relevant fluents
            0.03 seconds computing LNF
            0.02 seconds building connectivity graph
            69.36 seconds searching, evaluating 18819 states, to a max depth of 0
            78.58 seconds total time

```

Como prueba, se ha puesto precipicio en la zona 2 y en la zona 11, lo que hace que el jugador se quede aislado y el resultado de la planificación es:



```
ff: parsing domain file
domain 'EJ3-DOMAIN' defined
... done.
ff: parsing problem file
problem 'EJ3-PROBLEM' defined
... done.

ff: goal can be simplified to FALSE. No plan will solve it
```

También se han quitado las zapatillas y el bikini del mapa, lo que ofrece el mismo resultado anterior.

4. Ejercicio 4:

- a) Para contabilizar los puntos del jugador se crea la función *totalPoints*. Para saber cuántos puntos se consiguen por dar un objeto a un personaje se crea otra función *pointsGiving character obj*, que indicará cuántos puntos se otorgan por dar un objeto a un personaje. En el efecto de la acción *GIVE* se añade una sentencia que incrementa *totalPoints* dependiendo del personaje al que se le ha dado el objeto y por el valor que tenía dicho objeto con dicho personaje.
- b) En las condiciones iniciales se indica que *totalPoints* comienza con 0 puntos. Teniendo en cuenta la tabla, se establece con la función *pointsGiving* cuantos puntos le corresponde a cada objeto y cada personaje. Se ha quitado como objetivo que todos los jugadores tengan que recibir un objeto y se ha añadido que los puntos totales deben ser mayores o igual que 50. Se ha quitado el objetivo de que todos los personajes tengan que recibir un objeto porque el programa *Metric-FF* devolvía una violación de segmento si se mantenían los dos objetivos.

Se ha intentado establecer como objetivo (para este ejercicio y para el ejercicio 5) que todos los jugadores tengan que recibir, al menos, un objeto y que los puntos totales deben ser mayores o igual que 50, pero se obtiene un error de “violación de segmento”:

```
ff: parsing domain file
domain 'EJ4-DOMAIN' defined
... done.
ff: parsing problem file
problem 'EJ4-PROBLEM' defined
... done.

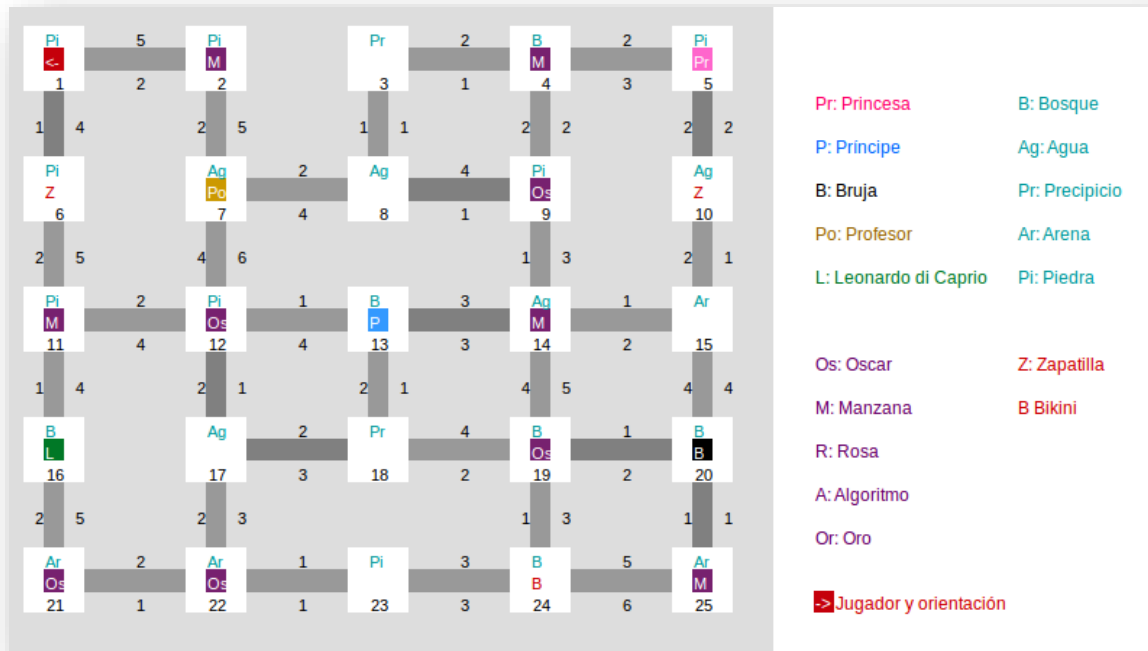
no optimization required. skipping criterion.

./launch.sh: línea 8: 3856 Violación de segmento ('core' generado) /home/fran/Escritorio/Universidad/TSI/Prácticas/Práctica\ 3/Sesión\ 1/Metric
-FF\ -\ Nuevo/ff -p ./ -o $1 -f $2
```

El código que origina este error es:

```
284 ; Objetivo final del problema
285 (:goal
286
287 ; El objetivo es que cada personaje tenga un objeto sin importar el
288 ; que sea
289 (and
290
291 (exists (?o1 ?o2 ?o3 ?o4 ?o5 - obj)
292 (and
293 (has princess ?o1)
294 (has prince ?o2)
295 (has witch ?o3)
296 (has professor ?o4)
297 (has leo ?o5)
298 )
299 )
300
301 (>= (totalPoints) 50)
302
303 )
```

Para cumplir con el requisito del apartado B que indica que se alcancen los 50 puntos solo con objetos de tipo oscar y manzana, el mapa del mundo queda:



El resultado de ejecutar la planificación es (sin optimización porque no es capaz de ofrecer un resultado en un tiempo razonable):

```
ff: found legal plan as follows
step    0: TURN_LEFT P
        1: GO P Z1 Z6 SOUTH
        2: GET P SHOE Z6
        3: GO P Z6 Z11 SOUTH
        4: RELEASE P SHOE Z11
        5: GET P APPLE Z11
        6: PUT_BACKPACK P APPLE
        7: GET P SHOE Z11
        8: GO P Z11 Z16 SOUTH
        9: GIVE P LEO APPLE Z16
       10: GO P Z16 Z21 SOUTH
       11: TURN_LEFT P
       12: TURN_LEFT P
       13: GET P OSCAR Z21
       14: GO P Z21 Z16 NORTH
       15: GIVE P LEO OSCAR Z16
       16: GO P Z16 Z11 NORTH
       17: TURN_RIGHT P
       18: GO P Z11 Z12 EAST
       19: TURN_LEFT P
       20: TURN_LEFT P
       21: GET P OSCAR Z12
       22: GO P Z12 Z11 WEST
       23: TURN_LEFT P
       24: GO P Z11 Z16 SOUTH
       25: GIVE P LEO OSCAR Z16
       26: GO P Z16 Z21 SOUTH
       27: TURN_LEFT P
       28: GO P Z21 Z22 EAST
       29: GO P Z22 Z23 EAST
       30: GO P Z23 Z24 EAST
       31: GO P Z24 Z25 EAST
       32: TURN_LEFT P
       33: GET P APPLE Z25
       34: GO P Z25 Z20 NORTH
       35: GIVE P WITCH APPLE Z20
       36: TURN_LEFT P
       37: TURN_LEFT P
       38: GO P Z20 Z25 SOUTH
       39: TURN_RIGHT P
       40: GO P Z25 Z24 WEST
       41: GO P Z24 Z23 WEST
       42: GO P Z23 Z22 WEST
       43: GET P OSCAR Z22
       44: GO P Z22 Z21 WEST
       45: TURN_RIGHT P
       46: GO P Z21 Z16 NORTH
       47: GIVE P LEO OSCAR Z16
       48: TURN_RIGHT P
       49: TURN_RIGHT P
       50: GO P Z16 Z21 SOUTH
```

```

50: GO P Z16 Z21 SOUTH
51: TURN_LEFT P
52: GO P Z21 Z22 EAST
53: GO P Z22 Z23 EAST
54: GO P Z23 Z24 EAST
55: GET P BIKINI Z24
56: TURN_LEFT P
57: GO P Z24 Z19 NORTH
58: GO P Z19 Z14 NORTH
59: TURN_RIGHT P
60: RELEASE P BIKINI Z14
61: GET P APPLE Z14
62: GO P Z14 Z15 EAST
63: TURN_RIGHT P
64: GO P Z15 Z20 SOUTH
65: GIVE P WITCH APPLE Z20
    Coste Total: 0.00

time spent:  0.00 seconds instantiating 1241 easy, 63 hard action templates
            0.00 seconds reachability analysis, yielding 203 facts and 262 actions
            0.00 seconds creating final representation with 177 relevant facts, 2 relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.03 seconds searching, evaluating 844 states, to a max depth of 18
            0.03 seconds total time

```

Para comprobar que ha obtenido, al menos, 50 puntos, se suman los puntos que se otorgan al dar a cada personaje un objeto:

- *9: GIVE P LEO APPLE Z16* → 3
- *15: GIVE P LEO OSCAR Z16* → 10
- *25: GIVE P LEO OSCAR Z16* → 10
- *35: GIVE P WITCH APPLE Z20* → 10
- *47: GIVE P LEO OSCAR Z16* → 10
- *65: GIVE P WITCH APPLE Z20* → 10
- *Total* → 53

5. Ejercicio 5:

- a) Para disponer de capacidad en la mochila se han añadido dos funciones:
- *countBackpack*: es un contador que indica la cantidad de objetos que hay en la mochila.
 - *backpackCapacity*: contiene la capacidad máxima que puede albergar la mochila.
- En las acciones de desplazamiento no se tiene en cuenta nada de la capacidad de la mochila, por lo que no se modifican.
- Las acciones que se modifican son:
- *PUT_BACKPACK*:
Se cambia la precondition que indicaba que si existía algún objeto en ella no se podía añadir otro y se sustituye por una que indica que se puede meter en la mochila si no existe ya ese mismo objeto. Se hace así porque no se puede saber (con lo implementado hasta ahora) si al sacar un objeto, queda otro igual dentro.
Se añade otra precondition que establece que *countBackpack* debe ser menor que *backpackCapacity*, para no poder añadir más objetos de los que se debe.
Un efecto adicional es que se incrementa en 1 la cantidad de objetos que contiene la mochila.
 - *POP_BACKPACK*:
No tiene preconditiones nuevas.
El efecto nuevo que se añade es que disminuye en 1 la cantidad de objetos que tiene.
- b) Para llevar la cuenta de los objetos, solo es necesario inicializar dos funciones:
- *backpackCapacity*: para indicar la capacidad de la mochila.
 - *countBackpack*: que se debe inicializar a 0 porque no existe ningún objeto en la mochila al inicio.

Para que se pueda ver un correcto funcionamiento de esta mejora, se ha optado por añadir los cinco objetos en la zona 1 y otros cinco objetos en la zona 15, con el fin de ver como aumenta o disminuye la posibilidad de meter objetos en la mochila si se cambia la capacidad.

Todos los resultados que se muestran son sin optimización.

Así, una ejecución de la planificación con capacidad 10 sería:

```
ff: found legal plan as follows
step  0: TURN_LEFT P
      1: GET P APPLE Z1
      2: PUT_BACKPACK P APPLE
      3: GET P ALGORITHM Z1
      4: PUT_BACKPACK P ALGORITHM
      5: GET P GOLD Z1
      6: PUT_BACKPACK P GOLD
      7: GET P OSCAR Z1
      8: GO P Z1 Z6 SOUTH
      9: PUT_BACKPACK P OSCAR
     10: GET P SHOE Z6
     11: GO P Z6 Z11 SOUTH
     12: GO P Z11 Z16 SOUTH
     13: GIVE P LEO GOLD Z16
     14: POP_BACKPACK P ALGORITHM
     15: GIVE P LEO ALGORITHM Z16
     16: POP_BACKPACK P APPLE
     17: GIVE P LEO APPLE Z16
     18: POP_BACKPACK P OSCAR
     19: GIVE P LEO OSCAR Z16
     20: GO P Z16 Z21 SOUTH
     21: TURN_LEFT P
     22: GO P Z21 Z22 EAST
     23: GO P Z22 Z23 EAST
     24: GO P Z23 Z24 EAST
     25: GO P Z24 Z25 EAST
     26: TURN_LEFT P
     27: GO P Z25 Z20 NORTH
     28: GO P Z20 Z15 NORTH
     29: TURN_LEFT P
     30: TURN_LEFT P
     31: GET P APPLE Z15
     32: GO P Z15 Z20 SOUTH
     33: GIVE P WITCH APPLE Z20
     34: GO P Z20 Z25 SOUTH
     35: TURN_RIGHT P
     36: GO P Z25 Z24 WEST
     37: TURN_LEFT P
     38: TURN_LEFT P
     39: GET P BIKINI Z24
     40: GO P Z24 Z25 EAST
     41: PUT_BACKPACK P BIKINI
     42: TURN_LEFT P
     43: GO P Z25 Z20 NORTH
     44: GO P Z20 Z15 NORTH
     45: GET P ROSE Z15
     46: GO P Z15 Z10 NORTH
     47: GO P Z10 Z5 NORTH
     48: GIVE P PRINCESS ROSE Z5
     49: TURN_LEFT P
     50: TURN_LEFT P
```

```

51: GO P Z5 Z10 SOUTH
52: GO P Z10 Z15 SOUTH
53: GET P ALGORITHM Z15
54: RELEASE P SHOE Z15
55: GET P GOLD Z15
56: TURN_RIGHT P
57: RELEASE P ALGORITHM Z15
58: GET P SHOE Z15
59: GO P Z15 Z14 WEST
60: GO P Z14 Z13 WEST
61: GIVE P PRINCE GOLD Z13
    Coste Total: 0.00

time spent:    0.00 seconds instantiating 1241 easy, 63 hard action templates
               0.01 seconds reachability analysis, yielding 293 facts and 421 actions
               0.00 seconds creating final representation with 288 relevant facts, 4 relevant fluents
               0.00 seconds computing LNF
               0.00 seconds building connectivity graph
               0.19 seconds searching, evaluating 6445 states, to a max depth of 15
               0.20 seconds total time

```

Para ver con más claridad las veces que se añade o se saca un objeto de la mochila se muestran los siguientes resultados:

- Capacidad 1:

```

2: PUT_BACKPACK P APPLE
6: POP_BACKPACK P APPLE
9: PUT_BACKPACK P SHOE
73: POP_BACKPACK P SHOE
76: PUT_BACKPACK P BIKINI

```

- Capacidad 2:

```

2: PUT_BACKPACK P APPLE
4: PUT_BACKPACK P ALGORITHM

```

- Capacidad 3:

```

2: PUT_BACKPACK P APPLE
4: PUT_BACKPACK P ALGORITHM
6: PUT_BACKPACK P GOLD
15: POP_BACKPACK P GOLD

```

- Capacidad 4:

```
2: PUT_BACKPACK P APPLE
4: PUT_BACKPACK P ALGORITHM
6: PUT_BACKPACK P GOLD
9: PUT_BACKPACK P OSCAR
14: POP_BACKPACK P ALGORITHM
16: POP_BACKPACK P APPLE
18: POP_BACKPACK P OSCAR
41: PUT_BACKPACK P BIKINI
```

A partir de la capacidad 4, el resultado es siempre el mismo.