

Técnicas de los Sistemas Inteligentes (2015-2016)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Práctica 3: Primera Parte

Francisco Carrillo Pérez

25 de junio de 2016

Índice

1. Ejercicio 1	3
1.1. Apartado a	3
1.2. Apartado b	3
1.3. Apartado c	4
1.4. Apartado d	6
2. Ejercicio 2	9
2.1. Apartado a	9
2.2. Apartado b	9
3. Ejercicio 3	11
3.1. Apartado a	11
3.2. Apartado b	12
3.3. Apartado c	12
4. Ejercicio 4	15
4.1. Apartado a	15
4.2. Apartado b	16
5. Ejercicio 5	18
5.1. Apartado a	18
5.2. Apartado b	19

Índice de figuras

1.1. Primera parte del plan encontrado	8
1.2. Primera parte del plan encontrado	9
2.1. Primera parte del plan encontrado para el segundo ejercicio	10
2.2. Segunda parte del plan encontrado para el segundo ejercicio	10
3.1. Primera parte del plan encontrado para el tercer ejercicio	14
3.2. Segunda parte del plan encontrado para el tercer ejercicio	14
3.3. Tercera parte del plan encontrado para el tercer ejercicio	15
4.1. Primera parte del plan encontrado para el cuarto ejercicio	17
4.2. Segunda parte del plan encontrado para el cuarto ejercicio	17
4.3. Tercera parte del plan encontrado para el cuarto ejercicio	18
5.1. Primera parte del plan encontrado para el quinto ejercicio	19
5.2. Segunda parte del plan encontrado para el quinto ejercicio	19
5.3. Tercera parte del plan encontrado para el quinto ejercicio	20

Introducción

En el siguiente documento se va a proceder a la explicación a las decisiones de diseño que se han tomado a la hora de realizar 5 ejercicios de los 6 propuestos, además de mostrar los distintos planes que se han conseguido para cada dominio y problema.

1. Ejercicio 1

1.1. Apartado a

Para la representación en el dominio de los objetos en el mundo se ha optado por tener 4 tipos:

- **robot**: Para representar al robot.
- **zona**: Para representar las distintas zonas dentro del mundo.
- **objeto**: Para representar los distintos objetos que hay en el mundo.
- **personaje**: Para representar a los distintos personajes que hay en el mundo.

Podemos observarlo de la siguiente forma en el archivo del dominio:

```
(:types robot zona objeto personaje)
```

1.2. Apartado b

Los predicados que se han descrito son los siguientes:

- **(at ?x - robot ?y - zona)**: Para representar en que zona se encuentra el robot.
- **(at1 ?x - objeto ?y - zona)**: Para representar en que zona se encuentra un objeto.
- **(at2 ?x - personaje ?y - zona)**: Para representar en que zona se encuentra un personaje.
- **(conectada ?x - zona ?y - zona)**: Para representar que dos zonas se encuentran conectadas.
- **(cogido ?x - objeto)**: Para representar que el robot tiene cogido un determinado objeto.
- **(tiene ?x - objeto ?y - personaje)**: Para representar que un determinado personaje tiene un determinado objeto.
- **(manovacia)**: Para representar que el robot no tiene ningún objeto cogido.

Podemos observarlo de la siguiente forma en el archivo del dominio:

```
;Declaracion de los predicados del mundo de Belkan
(:predicates
  (at ?x - robot ?y - zona)
  (at1 ?x - objeto ?y - zona)
  (at2 ?x - personaje ?y - zona)
  (conectada ?x - zona ?y - zona)
  (cogido ?x - objeto)
  (tiene ?x - objeto ?y - personaje)
  (manovacia)
)
```

1.3. Apartado c

Las acciones se han representado de la siguiente forma:

Ir de una zona a otra

```
;Acción de movimiento del robot
(:action mover-robot
  :parameters (?r - robot ?z1 - zona ?z2 - zona)
  :precondition (and (not (at ?r ?z2))(at ?r ?z1)(conectada ?z1 ?z2))
  :effect (and (not (at ?r ?z1))(at ?r ?z2))
)
```

Así tenemos:

- **Parámetros:** El robot, la zona donde estamos y la zona a donde deseamos desplazarnos.
- **Precondiciones:** Que el robot no se encuentre en la zona a donde deseamos desplazarnos, que el robot se encuentre en la primera zona, y que ambas zonas estén conectadas, para poder desplazarnos de una a otra.
- **Efecto:** Que el robot no se encuentre en la primera zona donde estaba, y que ahora se encuentre en la segunda zona.

Coger un objeto

```
;Acción de coger un objeto
(:action coger-objeto
  :parameters (?r - robot ?obj - objeto ?obj1 - objeto ?z - zona)
  :precondition (and (at ?r ?z) (at1 ?obj ?z)(manovacia))
  :effect (and (not (manovacia))(cogido ?obj))
)
```

Así tenemos:

- **Parámetros:** El robot, el objeto que queremos coger y la zona donde nos encontramos y se encuentra el objeto.
- **Precondiciones:** Que el robot y el objeto se encuentren en la misma zona, y que tengamos la mano vacía.
- **Efecto:** El robot deja de tener la mano vacía y pasa a tener cogido el objeto.

Dejar un objeto

```
;Acción de dejar un objeto
(:action dejar-objeto
  :parameters (?r - robot ?obj - objeto ?z - zona)
  :precondition (and (at ?r ?z)(cogido ?obj))
  :effect (and (not (cogido ?obj))(at1 ?obj ?z)(manovacia))
)
```

Así tenemos:

- **Parámetros:** El robot, el objeto que queremos dejar y la zona donde nos encontramos y queremos dejar el objeto.
- **Precondiciones:** Que el robot y el objeto se encuentren en la misma zona, y que tengamos el objeto cogido.
- **Efecto:** El robot pasa a tener la mano vacía, deja cogido el objeto y se inserta que el objeto se encuentre en esa zona.

Entregar un objeto a un personaje

```
;Entregar un objeto a un personaje
(:action entregar-objeto
  :parameters(?r - robot ?obj - objeto ?z - zona ?p - personaje)
  :precondition (and (not (tiene ?obj ?p))(at ?r ?z)(at2 ?p ?z)(cogido ?obj))
  :effect (and (not (cogido ?obj))(tiene ?obj ?p)(manovacia))
)
```

Así tenemos:

- **Parámetros:** El robot, el objeto que queremos entregar . la zona donde nos encontramos y el personaje al que queremos entregar el objeto.
- **Precondiciones:** Que el personaje no tenga objeto, que el robot y el personaje estén en la misma zona y que tenga cogido el objeto.

- **Efecto:** El robot ya no tiene cogido el objeto, el personaje tiene el objeto y tiene la mano vacía.

1.4. Apartado d

El archivo de problema es el siguiente:

```
(define (problem Problema1)

(:domain LosMundosDeBelkan)

(:OBJECTS Z1 Z2 Z3 Z4 Z5 Z6 Z7 Z8 Z9 Z10 Z11 Z12 Z13 Z14 Z15 Z16
Z17 Z18 Z19 Z20 Z21 Z22 Z23 Z24 Z25 - zona ROBOT - robot
OSCAR MANZANA ROSA ALGORITMO ORO - objeto PRINCESA PRINCIPE BRUJA
PROFESOR LEONARDO - personaje)

(:INIT (CONECTADA Z1 Z2)
(CONECTADA Z2 Z1)
(CONECTADA Z1 Z6)
(CONECTADA Z6 Z1)
(CONECTADA Z2 Z3)
(CONECTADA Z3 Z2)
(CONECTADA Z2 Z7)
(CONECTADA Z7 Z2)
(CONECTADA Z3 Z4)
(CONECTADA Z4 Z3)
(CONECTADA Z3 Z8)
(CONECTADA Z8 Z3)
(CONECTADA Z4 Z5)
(CONECTADA Z5 Z4)
(CONECTADA Z4 Z9)
(CONECTADA Z9 Z4)
(CONECTADA Z5 Z10)
(CONECTADA Z10 Z5)
(CONECTADA Z6 Z7)
(CONECTADA Z7 Z6)
(CONECTADA Z6 Z11)
(CONECTADA Z11 Z6)
(CONECTADA Z7 Z8)
(CONECTADA Z8 Z7)
(CONECTADA Z7 Z12)
(CONECTADA Z12 Z7)
(CONECTADA Z8 Z9)
```

(CONECTADA Z9 Z8)
(CONECTADA Z8 Z13)
(CONECTADA Z13 Z8)
(CONECTADA Z9 Z10)
(CONECTADA Z10 Z9)
(CONECTADA Z9 Z14)
(CONECTADA Z14 Z9)
(CONECTADA Z10 Z15)
(CONECTADA Z15 Z10)
(CONECTADA Z11 Z12)
(CONECTADA Z12 Z11)
(CONECTADA Z11 Z16)
(CONECTADA Z16 Z11)
(CONECTADA Z12 Z13)
(CONECTADA Z13 Z12)
(CONECTADA Z12 Z17)
(CONECTADA Z17 Z12)
(CONECTADA Z13 Z14)
(CONECTADA Z14 Z13)
(CONECTADA Z13 Z18)
(CONECTADA Z18 Z13)
(CONECTADA Z14 Z15)
(CONECTADA Z15 Z14)
(CONECTADA Z14 Z19)
(CONECTADA Z19 Z14)
(CONECTADA Z15 Z20)
(CONECTADA Z20 Z15)
(CONECTADA Z16 Z17)
(CONECTADA Z17 Z16)
(CONECTADA Z16 Z21)
(CONECTADA Z21 Z16)
(CONECTADA Z17 Z18)
(CONECTADA Z18 Z17)
(CONECTADA Z17 Z22)
(CONECTADA Z22 Z17)
(CONECTADA Z18 Z19)
(CONECTADA Z19 Z18)
(CONECTADA Z19 Z20)
(CONECTADA Z20 Z19)
(CONECTADA Z20 Z25)
(CONECTADA Z25 Z20)
(CONECTADA Z21 Z22)
(CONECTADA Z22 Z21)
(CONECTADA Z22 Z23)

```

(CONECTADA Z23 Z22)
(CONECTADA Z24 Z25)
(CONECTADA Z25 Z24)
(AT1 ORO Z2)
(AT2 PRINCIPE Z4)
(AT2 LEONARDO Z6)
(AT2 BRUJA Z7)
(AT1 OSCAR Z10)
(AT1 MANZANA Z11)
(AT2 PRINCESA Z13)
(AT1 ROSA Z14)
(AT2 PROFESOR Z21)
(AT1 ALGORITMO Z25)
(AT ROBOT Z12)
(MANOVACIA)
)

```

```

(:goal (AND (TIENE ORO PRINCIPE)(TIENE OSCAR LEONARDO)(TIENE MANZANA BRUJA)
(TIENE ALGORITMO PROFESOR)(TIENE ROSA PRINCESA)))

)

```

Y en las siguientes imágenes podemos observar como encuentra un plan:

```

Archivo  Editar  Ver  Bookmarks  Preferencias  Ayuda
1
0 [1]
ff: found legal plan as follows
step  0: MOVER-ROBOT ROBOT Z12 Z11
      1: COGER-OBJETO ROBOT MANZANA Z11
      2: MOVER-ROBOT ROBOT Z11 Z12
      3: MOVER-ROBOT ROBOT Z12 Z7
      4: ENTREGAR-OBJETO ROBOT MANZANA Z7 BRUJA
      5: MOVER-ROBOT ROBOT Z7 Z6
      6: MOVER-ROBOT ROBOT Z6 Z11
      7: MOVER-ROBOT ROBOT Z11 Z12
      8: MOVER-ROBOT ROBOT Z12 Z13
      9: MOVER-ROBOT ROBOT Z13 Z14
     10: COGER-OBJETO ROBOT ROSA Z14
     11: MOVER-ROBOT ROBOT Z14 Z13
     12: ENTREGAR-OBJETO ROBOT ROSA Z13 PRINCESA
     13: MOVER-ROBOT ROBOT Z13 Z12
     14: MOVER-ROBOT ROBOT Z12 Z7
     15: MOVER-ROBOT ROBOT Z7 Z2
     16: COGER-OBJETO ROBOT ORO Z2
     17: MOVER-ROBOT ROBOT Z2 Z3
     18: MOVER-ROBOT ROBOT Z3 Z4
     19: ENTREGAR-OBJETO ROBOT ORO Z4 PRINCIPE
     20: MOVER-ROBOT ROBOT Z4 Z5

```

Figura 1.1: Primera parte del plan encontrado

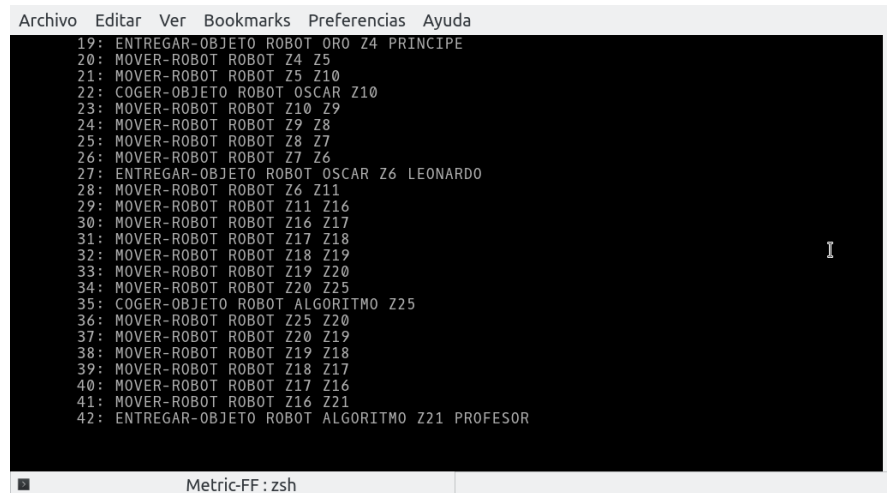


Figura 1.2: Primera parte del plan encontrado

2. Ejercicio 2

2.1. Apartado a

Para añadir esta característica se ha usado una función coste-fuel para indicar el fuel que se ha gastado, de la siguiente forma:

```
(:functions
  (coste-fuel) ;representa el fuel del robot
)
```

No se ha añadido ningún predicado nuevo, pero con cada movimiento del robot, se consume fuel, por lo que se ha añadido un incremento de fuel como efecto de la acción **mover-robot**:

```
;Acción de movimiento del robot
(:action mover-robot
  :parameters (?r - robot ?z1 - zona ?z2 - zona)
  :precondition (and (not (at ?r ?z2))(at ?r ?z1)(conectada ?z1 ?z2))
  :effect (and (not (at ?r ?z1))(at ?r ?z2)(increase (coste-fuel) 1))
)
```

2.2. Apartado b

El dominio del problema se ha modificado en los siguientes aspectos:

- Se ha añadido en `:init` un nuevo predicado que nos dice que el coste del fuel al comienzo es 0.

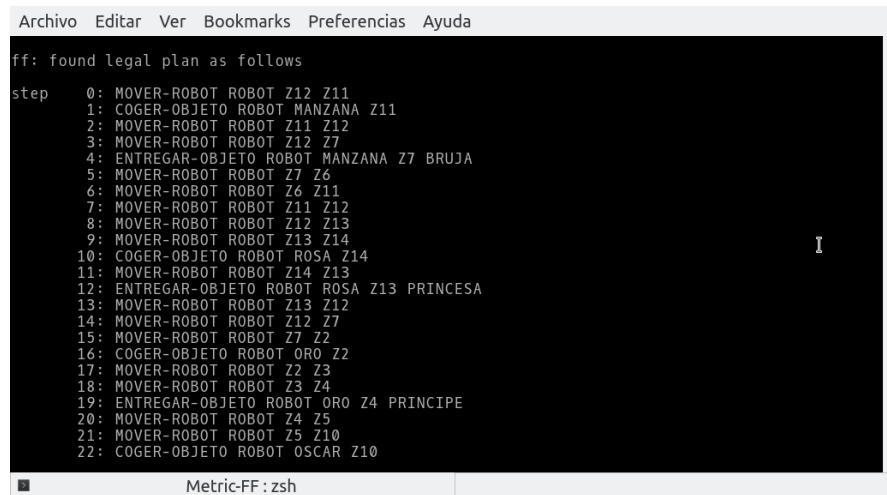
```
(= (coste-fuel) 0)
```

- Se ha añadido que queremos que el plan minimice el coste de fuel de la siguiente forma:

```
(:metric minimize (coste-fuel))
```

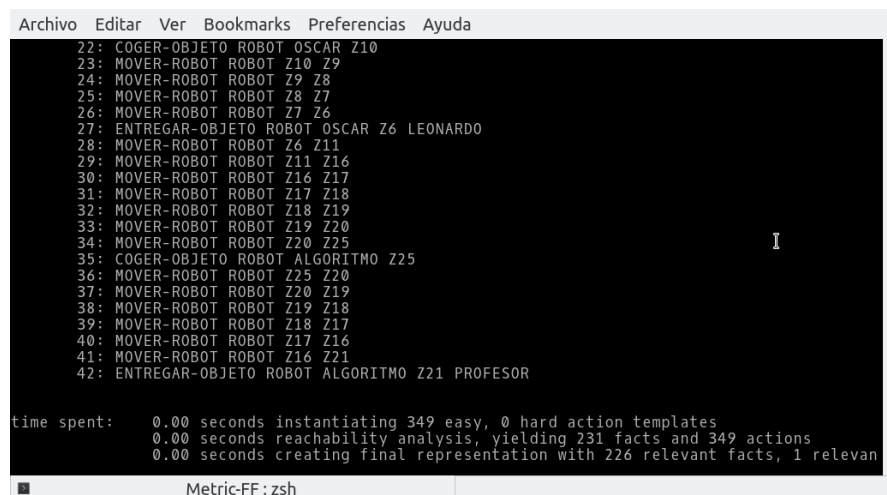
El resto es exactamente igual que el apartado d del ejercicio 1.

Podemos observar su correcto funcionamiento en las siguientes imágenes:



```
Archivo  Editar  Ver  Bookmarks  Preferencias  Ayuda
ff: found legal plan as follows
step    0: MOVER-ROBOT ROBOT Z12 Z11
        1: COGER-OBJETO ROBOT MANZANA Z11
        2: MOVER-ROBOT ROBOT Z11 Z12
        3: MOVER-ROBOT ROBOT Z12 Z7
        4: ENTREGAR-OBJETO ROBOT MANZANA Z7 BRUJA
        5: MOVER-ROBOT ROBOT Z7 Z6
        6: MOVER-ROBOT ROBOT Z6 Z11
        7: MOVER-ROBOT ROBOT Z11 Z12
        8: MOVER-ROBOT ROBOT Z12 Z13
        9: MOVER-ROBOT ROBOT Z13 Z14
       10: COGER-OBJETO ROBOT ROSA Z14
       11: MOVER-ROBOT ROBOT Z14 Z13
       12: ENTREGAR-OBJETO ROBOT ROSA Z13 PRINCESA
       13: MOVER-ROBOT ROBOT Z13 Z12
       14: MOVER-ROBOT ROBOT Z12 Z7
       15: MOVER-ROBOT ROBOT Z7 Z2
       16: COGER-OBJETO ROBOT ORO Z2
       17: MOVER-ROBOT ROBOT Z2 Z3
       18: MOVER-ROBOT ROBOT Z3 Z4
       19: ENTREGAR-OBJETO ROBOT ORO Z4 PRINCIPE
       20: MOVER-ROBOT ROBOT Z4 Z5
       21: MOVER-ROBOT ROBOT Z5 Z10
       22: COGER-OBJETO ROBOT OSCAR Z10
```

Figura 2.1: Primera parte del plan encontrado para el segundo ejercicio



```
Archivo  Editar  Ver  Bookmarks  Preferencias  Ayuda
22: COGER-OBJETO ROBOT OSCAR Z10
23: MOVER-ROBOT ROBOT Z10 Z9
24: MOVER-ROBOT ROBOT Z9 Z8
25: MOVER-ROBOT ROBOT Z8 Z7
26: MOVER-ROBOT ROBOT Z7 Z6
27: ENTREGAR-OBJETO ROBOT OSCAR Z6 LEONARDO
28: MOVER-ROBOT ROBOT Z6 Z11
29: MOVER-ROBOT ROBOT Z11 Z16
30: MOVER-ROBOT ROBOT Z16 Z17
31: MOVER-ROBOT ROBOT Z17 Z18
32: MOVER-ROBOT ROBOT Z18 Z19
33: MOVER-ROBOT ROBOT Z19 Z20
34: MOVER-ROBOT ROBOT Z20 Z25
35: COGER-OBJETO ROBOT ALGORITMO Z25
36: MOVER-ROBOT ROBOT Z25 Z20
37: MOVER-ROBOT ROBOT Z20 Z19
38: MOVER-ROBOT ROBOT Z19 Z18
39: MOVER-ROBOT ROBOT Z18 Z17
40: MOVER-ROBOT ROBOT Z17 Z16
41: MOVER-ROBOT ROBOT Z16 Z21
42: ENTREGAR-OBJETO ROBOT ALGORITMO Z21 PROFESOR

time spent:  0.00 seconds instantiating 349 easy, 0 hard action templates
            0.00 seconds reachability analysis, yielding 231 facts and 349 actions
            0.00 seconds creating final representation with 226 relevant facts, 1 relevant
```

Figura 2.2: Segunda parte del plan encontrado para el segundo ejercicio

3. Ejercicio 3

3.1. Apartado a

Para tener en consideración las restricciones se han realizado los siguientes cambios en el dominio anteriormente usado:

1. Se ha añadido un nuevo type llamado **tipo** para determinar el tipo de la zona, y un nuevo type llamado mochila, para trabajar con esta misma.

```
(:types robot zona objeto personaje tipo mochila)
```

2. Se ha añadido una nueva función para determinar el coste de una zona según el tipo de la misma.

```
(coste-zona ?z - tipo) ;coste de una zona
```

3. Se han añadido los siguientes predicados nuevos:

(mochilavacia): indica que la mochila se encuentra vacía.

(contiene ?x - objeto): indica que la mochila contiene un determinado objeto.

(clase ?x - zona ?y - tipo): indica de que tipo es una determinada zona.

(permite-avanzar ?x - objeto ?y - tipo): indica para un determinado tipo de zona que objeto debemos tener para poder avanzar por esa zona.

(sin-objeto-necesario ?x - tipo): indica que un tipo de zona no necesita ningún objeto para poder avanzar por ella.

4. También se han realizado las siguientes modificaciones en la acción de mover-robot:

-Tenga cogido o en la mochila el objeto si quiere pasar por una zona que lo necesite

-Se compruebe si se necesita objeto para pasar por el tipo de la zona que quiere.

-La diferencia entre el coste de fuel que tiene y el coste de la zona sea mayor o igual a 0.

-Ahora se decrementa el fuel según el coste de la zona.

;Acción de movimiento del robot

```
(:action mover-robot
```

```
  :parameters (?r - robot ?z1 - zona ?z2 - zona ?t - tipo ?obj - objeto)
```

```
  :precondition (and (not (at ?r ?z2))(at ?r ?z1)(conectada ?z1 ?z2)
```

```
    (or(cogido ?obj)(contiene ?obj))(or (permite-avanzar ?obj ?t)
```

```
    (sin-objeto-necesario ?t))(>=(-(coste-fuel)(coste-zona ?t))0))
```

```
  :effect (and (not (at ?r ?z1))(at ?r ?z2)
```

```
    (decrease (coste-fuel)(coste-zona ?t)))
```

```
)
```

3.2. Apartado b

Para ello se han añadido las siguientes acciones:

1. meter-mochila

```
;Guardar un objeto en la mochila
(:action meter-mochila
  :parameters(?r - robot ?obj - objeto)
  :precondition(and (cogido ?obj)(mochilavacia))
  :effect(and (not (cogido ?obj))(not (mochilavacia))(manovacia)(contiene ?obj))
)
```

Para poder meter un objeto en la mochila, la mochila debe estar vacía, y debmos tener el objeto cogido.

2. sacar-mochila

```
;Sacar un objeto de la mochila
(:action sacar-mochila
  :parameters(?r - robot ?obj - objeto)
  :precondition(and (contiene ?obj)(manovacia))
  :effect(and (not (contiene ?obj))(not (manovacia))(mochilavacia)(cogido ?obj))
)
```

Para poder sacar el objeto de la mochila debe tenerlo, por supuesto, y además debe tener la mano vacía.

3.3. Apartado c

Las modificaciones que se han realizado al archivo de problema anterior son las siguientes:

1. Se han añadido los objetos zapatilla y bikini, además de los tipos de zonas.

```
OSCAR MANZANA ROSA ALGORITMO ORO ZAPATILLA BIKINI - objeto
BOSQUE AGUA PRECIPICIO ARENA PIEDRA - tipo
```

2. Se han añadido los siguientes predicados nuevos y modificado el de coste-fuel y se ha puesto a valor 100:

```
(MOCHILAVACIA)
(CLASE Z1 PRECIPICIO)
(CLASE Z2 PIEDRA)
(CLASE Z3 PIEDRA)
(CLASE Z4 PIEDRA)
(CLASE Z5 PIEDRA)
```

```

(CLASE Z6 PIEDRA)
(CLASE Z7 BOSQUE)
(CLASE Z8 BOSQUE)
(CLASE Z9 BOSQUE)
(CLASE Z10 AGUA)
(CLASE Z11 AGUA)
(CLASE Z12 AGUA)
(CLASE Z13 AGUA)
(CLASE Z14 AGUA)
(CLASE Z15 AGUA)
(CLASE Z16 ARENA)
(CLASE Z17 ARENA)
(CLASE Z18 ARENA)
(CLASE Z19 ARENA)
(CLASE Z20 ARENA)
(CLASE Z21 BOSQUE)
(CLASE Z22 BOSQUE)
(CLASE Z23 PRECIPICIO)
(CLASE Z24 BOSQUE)
(CLASE Z25 BOSQUE)
(PERMITE-AVANZAR BIKINI AGUA)
(PERMITE-AVANZAR ZAPATILLA BOSQUE)
(SIN-OBJETO-NECESARIO PIEDRA)
(SIN-OBJETO-NECESARIO ARENA)
(AT1 BIKINI Z12)
(AT1 ZAPATILLA Z21)
(= (coste-zona PRECIPICIO) 10000)
(= (coste-zona AGUA) 1)
(= (coste-zona BOSQUE) 2)
(= (coste-zona ARENA) 1)
(= (coste-zona PIEDRA) 3)
(= (coste-fuel) 100)

```

En las siguientes imágenes podemos observar como encuentra un plan:

```
Archivo Editar Ver Bookmarks Preferencias Ayuda
ff: found legal plan as follows
step  0: COGER-OBJETO ROBOT BIKINI BIKINI Z12
      1: MOVER-ROBOT ROBOT Z12 Z13 ARENA BIKINI
      2: MOVER-ROBOT ROBOT Z13 Z14 ARENA BIKINI
      3: METER-MOCHILA ROBOT BIKINI
      4: COGER-OBJETO ROBOT ROSA BIKINI Z14
      5: MOVER-ROBOT ROBOT Z14 Z13 ARENA ROSA
      6: DEJAR-OBJETO ROBOT ROSA Z13
      7: COGER-OBJETO ROBOT ROSA BIKINI Z13
      8: ENTREGAR-OBJETO ROBOT ROSA Z13 PRINCESA
      9: MOVER-ROBOT ROBOT Z13 Z8 PIEDRA BIKINI
     10: MOVER-ROBOT ROBOT Z8 Z3 PIEDRA BIKINI
     11: MOVER-ROBOT ROBOT Z3 Z2 PIEDRA BIKINI
     12: COGER-OBJETO ROBOT ORO BIKINI Z2
     13: MOVER-ROBOT ROBOT Z2 Z7 ARENA BIKINI
     14: MOVER-ROBOT ROBOT Z7 Z8 ARENA ORO
     15: DEJAR-OBJETO ROBOT ORO Z8
     16: MOVER-ROBOT ROBOT Z8 Z9 ARENA BIKINI
     17: MOVER-ROBOT ROBOT Z9 Z10 PIEDRA BIKINI
     18: COGER-OBJETO ROBOT OSCAR BIKINI Z10
     19: MOVER-ROBOT ROBOT Z10 Z9 ARENA BIKINI
     20: MOVER-ROBOT ROBOT Z9 Z8 ARENA BIKINI
     21: MOVER-ROBOT ROBOT Z8 Z7 ARENA BIKINI
     22: MOVER-ROBOT ROBOT Z7 Z6 PIEDRA BIKINI
Metric-FF: zsh
```

Figura 3.1: Primera parte del plan encontrado para el tercer ejercicio

```
Archivo Editar Ver Bookmarks Preferencias Ayuda
22: MOVER-ROBOT ROBOT Z7 Z6 PIEDRA BIKINI
23: DEJAR-OBJETO ROBOT OSCAR Z6
24: COGER-OBJETO ROBOT OSCAR BIKINI Z6
25: ENTREGAR-OBJETO ROBOT OSCAR Z6 LEONARDO
26: MOVER-ROBOT ROBOT Z6 Z7 ARENA BIKINI
27: MOVER-ROBOT ROBOT Z7 Z8 PIEDRA BIKINI
28: COGER-OBJETO ROBOT ORO BIKINI Z8
29: MOVER-ROBOT ROBOT Z8 Z3 PIEDRA ORO
30: MOVER-ROBOT ROBOT Z3 Z4 PIEDRA BIKINI
31: DEJAR-OBJETO ROBOT ORO Z4
32: COGER-OBJETO ROBOT ORO BIKINI Z4
33: ENTREGAR-OBJETO ROBOT ORO Z4 PRINCIPE
34: MOVER-ROBOT ROBOT Z4 Z9 ARENA BIKINI
35: MOVER-ROBOT ROBOT Z9 Z8 ARENA BIKINI
36: MOVER-ROBOT ROBOT Z8 Z7 ARENA BIKINI
37: MOVER-ROBOT ROBOT Z7 Z6 ARENA BIKINI
38: MOVER-ROBOT ROBOT Z6 Z11 ARENA BIKINI
39: COGER-OBJETO ROBOT MANZANA BIKINI Z11
40: MOVER-ROBOT ROBOT Z11 Z12 ARENA BIKINI
41: MOVER-ROBOT ROBOT Z12 Z7 ARENA MANZANA
42: DEJAR-OBJETO ROBOT MANZANA Z7
43: COGER-OBJETO ROBOT MANZANA BIKINI Z7
44: ENTREGAR-OBJETO ROBOT MANZANA Z7 BRUJA
45: MOVER-ROBOT ROBOT Z7 Z6 ARENA BIKINI
46: MOVER-ROBOT ROBOT Z6 Z11 ARENA BIKINI
47: MOVER-ROBOT ROBOT Z11 Z16 PIEDRA BIKINI
Metric-FF: zsh
```

Figura 3.2: Segunda parte del plan encontrado para el tercer ejercicio

```

Archivo  Editar  Ver  Bookmarks  Preferencias  Ayuda
46: MOVER-ROBOT ROBOT Z6 Z11 ARENA BIKINI
47: MOVER-ROBOT ROBOT Z11 Z16 PIEDRA BIKINI
48: MOVER-ROBOT ROBOT Z16 Z17 PIEDRA BIKINI
49: MOVER-ROBOT ROBOT Z17 Z18 PIEDRA BIKINI
50: MOVER-ROBOT ROBOT Z18 Z19 PIEDRA BIKINI
51: MOVER-ROBOT ROBOT Z19 Z20 PIEDRA BIKINI
52: MOVER-ROBOT ROBOT Z20 Z25 PIEDRA BIKINI
53: COGER-OBJETO ROBOT ALGORITMO BIKINI Z25
54: MOVER-ROBOT ROBOT Z25 Z20 ARENA ALGORITMO
55: MOVER-ROBOT ROBOT Z20 Z19 ARENA ALGORITMO
56: MOVER-ROBOT ROBOT Z19 Z18 ARENA ALGORITMO
57: MOVER-ROBOT ROBOT Z18 Z17 ARENA ALGORITMO
58: MOVER-ROBOT ROBOT Z17 Z16 ARENA ALGORITMO
59: MOVER-ROBOT ROBOT Z16 Z21 PIEDRA BIKINI
60: DEJAR-OBJETO ROBOT ALGORITMO Z21
61: COGER-OBJETO ROBOT ALGORITMO BIKINI Z21
62: ENTREGAR-OBJETO ROBOT ALGORITMO Z21 PROFESOR

time spent:  0.03 seconds instantiating 1449 easy, 2368 hard action templates
             0.00 seconds reachability analysis, yielding 311 facts and 3817 actions
             0.00 seconds creating final representation with 304 relevant facts, 1 relevan
t fluents
             0.01 seconds computing LNF
             0.00 seconds building connectivity graph
             38.52 seconds searching, evaluating 42135 states, to a max depth of 7

Metric-FF: zsh

```

Figura 3.3: Tercera parte del plan encontrado para el tercer ejercicio

4. Ejercicio 4

4.1. Apartado a

Se han añadido las siguientes funciones para poder contar los puntos de cada personaje y los puntos objetivo que queremos alcanzar:

(suma-puntos ?x - objeto ?y - personaje) ;Suma puntos de los personajes (puntos)

Con *suma-puntos* le indicamos que puntos le corresponden a la pareja de un objeto y un personaje determinado.

Con *puntos* determinamos la cantidad de puntos que queremos alcanzar como goal del problema.

Además la acción *entregar-objeto* se ha modificado para que incremente el número de puntos según el objeto que le damos al personaje, como podemos observar a continuación.

```

;Entregar un objeto a un personaje
(:action entregar-objeto
  :parameters(?r - robot ?obj - objeto ?z - zona ?p - personaje)
  :precondition (and (not (tiene ?obj ?p))(at ?r ?z)(at1 ?obj ?z)(at2 ?p ?z)
(cogido ?obj))
  :effect (and (not (cogido ?obj))(tiene ?obj ?p)(manovacia)
(increase (puntos) (suma-puntos ?obj ?p)))
)

```

4.2. Apartado b

Del problema del ejercicio 3 se han modificado los siguientes puntos para adaptarlo a estas nuevas condiciones:

1. Se han añadido todos los puntos correspondientes según el objeto y el personaje, según se nos indica en el enunciado del ejercicio.
2. Se inicializa el valor de puntos que se tiene al comienzo a 0.
3. Se cambia el goal por el número de puntos que deseamos alcanzar.

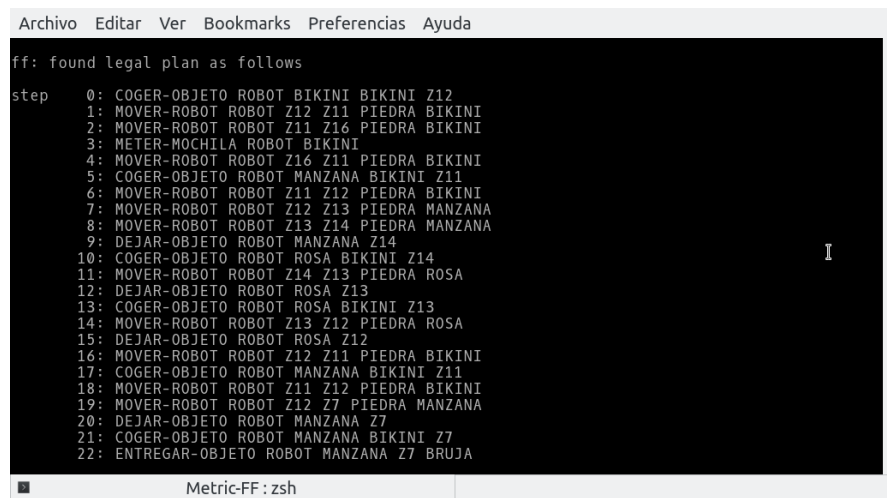
Se pueden observar los cambios a continuación:

```
(= (suma-puntos OSCAR LEONARDO) 10)
(= (suma-puntos OSCAR PRINCESA) 5)
(= (suma-puntos OSCAR BRUJA) 4)
(= (suma-puntos OSCAR PROFESOR) 3)
(= (suma-puntos OSCAR PRINCIPE) 1)
(= (suma-puntos ROSA LEONARDO) 1)
(= (suma-puntos ROSA PRINCESA) 10)
(= (suma-puntos ROSA BRUJA) 5)
(= (suma-puntos ROSA PROFESOR) 4)
(= (suma-puntos ROSA PRINCIPE) 3)
(= (suma-puntos MANZANA LEONARDO) 3)
(= (suma-puntos MANZANA PRINCESA) 1)
(= (suma-puntos MANZANA BRUJA) 10)
(= (suma-puntos MANZANA PROFESOR) 5)
(= (suma-puntos MANZANA PRINCIPE) 4)
(= (suma-puntos ALGORITMO LEONARDO) 4)
(= (suma-puntos ALGORITMO PRINCESA) 3)
(= (suma-puntos ALGORITMO BRUJA) 1)
(= (suma-puntos ALGORITMO PROFESOR) 10)
(= (suma-puntos ALGORITMO PRINCIPE) 5)
(= (suma-puntos ORO LEONARDO) 5)
(= (suma-puntos ORO PRINCESA) 4)
(= (suma-puntos ORO BRUJA) 3)
(= (suma-puntos ORO PROFESOR) 1)
(= (suma-puntos ORO PRINCIPE) 10)
(= (puntos) 0)

)

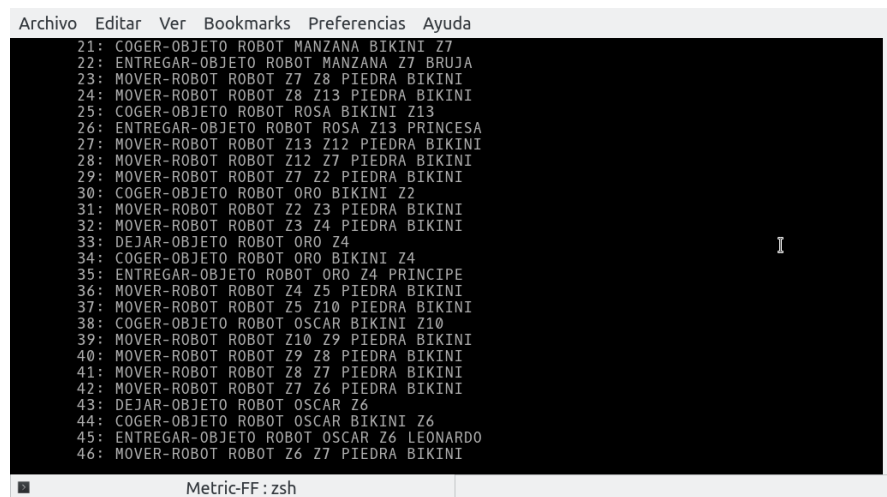
(:goal (= (puntos) 50))
)
```


En las siguientes imágenes podemos observar como encuentra un plan:



```
Archivo  Editar  Ver  Bookmarks  Preferencias  Ayuda
ff: found legal plan as follows
step    0: COGER-OBJETO ROBOT BIKINI BIKINI Z12
        1: MOVER-ROBOT ROBOT Z12 Z11 PIEDRA BIKINI
        2: MOVER-ROBOT ROBOT Z11 Z16 PIEDRA BIKINI
        3: METER-MOCHILA ROBOT BIKINI
        4: MOVER-ROBOT ROBOT Z16 Z11 PIEDRA BIKINI
        5: COGER-OBJETO ROBOT MANZANA BIKINI Z11
        6: MOVER-ROBOT ROBOT Z11 Z12 PIEDRA BIKINI
        7: MOVER-ROBOT ROBOT Z12 Z13 PIEDRA MANZANA
        8: MOVER-ROBOT ROBOT Z13 Z14 PIEDRA MANZANA
        9: DEJAR-OBJETO ROBOT MANZANA Z14
       10: COGER-OBJETO ROBOT ROSA BIKINI Z14
       11: MOVER-ROBOT ROBOT Z14 Z13 PIEDRA ROSA
       12: DEJAR-OBJETO ROBOT ROSA Z13
       13: COGER-OBJETO ROBOT ROSA BIKINI Z13
       14: MOVER-ROBOT ROBOT Z13 Z12 PIEDRA ROSA
       15: DEJAR-OBJETO ROBOT ROSA Z12
       16: MOVER-ROBOT ROBOT Z12 Z11 PIEDRA BIKINI
       17: COGER-OBJETO ROBOT MANZANA BIKINI Z11
       18: MOVER-ROBOT ROBOT Z11 Z12 PIEDRA BIKINI
       19: MOVER-ROBOT ROBOT Z12 Z7 PIEDRA MANZANA
       20: DEJAR-OBJETO ROBOT MANZANA Z7
       21: COGER-OBJETO ROBOT MANZANA BIKINI Z7
       22: ENTREGAR-OBJETO ROBOT MANZANA Z7 BRUJA
```

Figura 4.1: Primera parte del plan encontrado para el cuarto ejercicio



```
Archivo  Editar  Ver  Bookmarks  Preferencias  Ayuda
21: COGER-OBJETO ROBOT MANZANA BIKINI Z7
22: ENTREGAR-OBJETO ROBOT MANZANA Z7 BRUJA
23: MOVER-ROBOT ROBOT Z7 Z8 PIEDRA BIKINI
24: MOVER-ROBOT ROBOT Z8 Z13 PIEDRA BIKINI
25: COGER-OBJETO ROBOT ROSA BIKINI Z13
26: ENTREGAR-OBJETO ROBOT ROSA Z13 PRINCESA
27: MOVER-ROBOT ROBOT Z13 Z12 PIEDRA BIKINI
28: MOVER-ROBOT ROBOT Z12 Z7 PIEDRA BIKINI
29: MOVER-ROBOT ROBOT Z7 Z2 PIEDRA BIKINI
30: COGER-OBJETO ROBOT ORO BIKINI Z2
31: MOVER-ROBOT ROBOT Z2 Z3 PIEDRA BIKINI
32: MOVER-ROBOT ROBOT Z3 Z4 PIEDRA BIKINI
33: DEJAR-OBJETO ROBOT ORO Z4
34: COGER-OBJETO ROBOT ORO BIKINI Z4
35: ENTREGAR-OBJETO ROBOT ORO Z4 PRINCIPE
36: MOVER-ROBOT ROBOT Z4 Z5 PIEDRA BIKINI
37: MOVER-ROBOT ROBOT Z5 Z10 PIEDRA BIKINI
38: COGER-OBJETO ROBOT OSCAR BIKINI Z10
39: MOVER-ROBOT ROBOT Z10 Z9 PIEDRA BIKINI
40: MOVER-ROBOT ROBOT Z9 Z8 PIEDRA BIKINI
41: MOVER-ROBOT ROBOT Z8 Z7 PIEDRA BIKINI
42: MOVER-ROBOT ROBOT Z7 Z6 PIEDRA BIKINI
43: DEJAR-OBJETO ROBOT OSCAR Z6
44: COGER-OBJETO ROBOT OSCAR BIKINI Z6
45: ENTREGAR-OBJETO ROBOT OSCAR Z6 LEONARDO
46: MOVER-ROBOT ROBOT Z6 Z7 PIEDRA BIKINI
```

Figura 4.2: Segunda parte del plan encontrado para el cuarto ejercicio

```

Archivo  Editar  Ver  Bookmarks  Preferencias  Ayuda
46: MOVER-ROBOT ROBOT Z6 Z7 PIEDRA BIKINI
47: MOVER-ROBOT ROBOT Z7 Z12 PIEDRA BIKINI
48: COGER-OBJETO ROBOT ROSA BIKINI Z12
49: MOVER-ROBOT ROBOT Z12 Z7 PIEDRA ROSA
50: DEJAR-OBJETO ROBOT ROSA Z7
51: COGER-OBJETO ROBOT ROSA BIKINI Z7
52: ENTREGAR-OBJETO ROBOT ROSA Z7 BRUJA
53: MOVER-ROBOT ROBOT Z7 Z2 PIEDRA BIKINI
54: COGER-OBJETO ROBOT ORO BIKINI Z2
55: MOVER-ROBOT ROBOT Z2 Z1 PIEDRA BIKINI
56: MOVER-ROBOT ROBOT Z1 Z6 PIEDRA BIKINI
57: DEJAR-OBJETO ROBOT ORO Z6
58: COGER-OBJETO ROBOT ORO BIKINI Z6
59: ENTREGAR-OBJETO ROBOT ORO Z6 LEONARDO

time spent:  0.05 seconds instantiating 1449 easy, 2368 hard action templates
            0.00 seconds reachability analysis, yielding 311 facts and 3807 actions
            0.00 seconds creating final representation with 304 relevant facts, 3 relevant
t fluents
            0.01 seconds computing LNF
            0.02 seconds building connectivity graph
            9.78 seconds searching, evaluating 29546 states, to a max depth of 14
            9.86 seconds total time
-----
Metric-FF: zsh

```

Figura 4.3: Tercera parte del plan encontrado para el cuarto ejercicio

5. Ejercicio 5

5.1. Apartado a

Para trabajar con la capacidad de la mochila se han añadido dos nuevas funciones:

1. **(capacidad)**: que indica la capacidad de la mochila.
2. **(objetos-mochila)**: cantidad de objetos que llevo actualmente en la mochila.

En la acción de mover no se ha modificado, ya que consideraba si había algo en la mochila, y esto es indiferente al número de objetos que se encuentren en ella. En cambio, las acciones de cargar y sacar de la mochila se han modificado de la siguiente forma:

```

;Guardar un objeto en la mochila
(:action meter-mochila
  :parameters(?r - robot ?obj - objeto)
  :precondition(and (cogido ?obj)(>(- (capacidad)(objetos-mochila))0))
  :effect(and (not (cogido ?obj))(manovacia)(contiene ?obj)
    (increase (objetos-mochila)1))
)

;Sacar un objeto de la mochila
(:action sacar-mochila
  :parameters(?r - robot ?obj - objeto)
  :precondition(and (contiene ?obj)(manovacia))
  :effect(and (not (contiene ?obj))(not (manovacia))(cogido ?obj)
    (decrease (objetos-mochila)1))
)

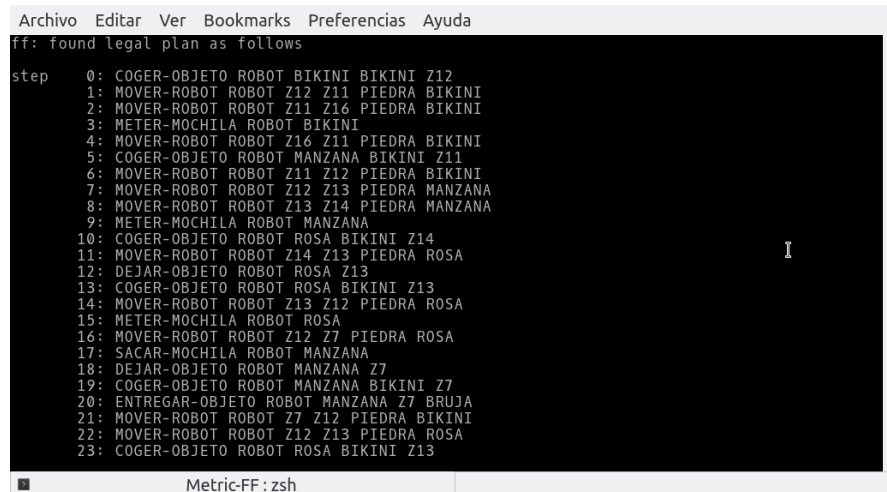
```

5.2. Apartado b

Se ha extendido el fichero de problema anterior añadiendo la capacidad de la mochila que queramos, y indicando que comenzamos con 0 objetos en la mochila:

```
= (capacidad) 3)
(= (objetos-mochila) 0)
```

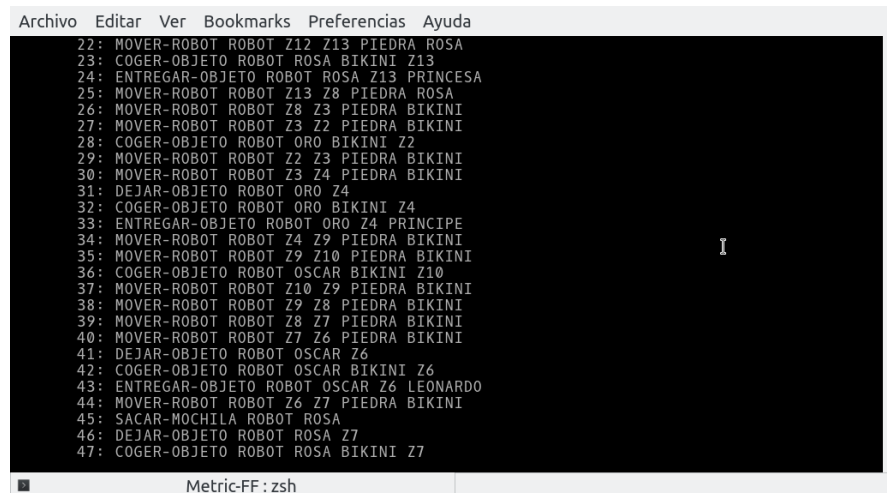
En las siguientes imágenes podemos observar como encuentra un plan:



```
Archivo Editar Ver Bookmarks Preferencias Ayuda
ff: found legal plan as follows

step  0: COGER-OBJETO ROBOT BIKINI BIKINI Z12
      1: MOVER-ROBOT ROBOT Z12 Z11 PIEDRA BIKINI
      2: MOVER-ROBOT ROBOT Z11 Z16 PIEDRA BIKINI
      3: METER-MOCHILA ROBOT BIKINI
      4: MOVER-ROBOT ROBOT Z16 Z11 PIEDRA BIKINI
      5: COGER-OBJETO ROBOT MANZANA BIKINI Z11
      6: MOVER-ROBOT ROBOT Z11 Z12 PIEDRA BIKINI
      7: MOVER-ROBOT ROBOT Z12 Z13 PIEDRA MANZANA
      8: MOVER-ROBOT ROBOT Z13 Z14 PIEDRA MANZANA
      9: METER-MOCHILA ROBOT MANZANA
     10: COGER-OBJETO ROBOT ROSA BIKINI Z14
     11: MOVER-ROBOT ROBOT Z14 Z13 PIEDRA ROSA
     12: DEJAR-OBJETO ROBOT ROSA Z13
     13: COGER-OBJETO ROBOT ROSA BIKINI Z13
     14: MOVER-ROBOT ROBOT Z13 Z12 PIEDRA ROSA
     15: METER-MOCHILA ROBOT ROSA
     16: MOVER-ROBOT ROBOT Z12 Z7 PIEDRA ROSA
     17: SACAR-MOCHILA ROBOT MANZANA
     18: DEJAR-OBJETO ROBOT MANZANA Z7
     19: COGER-OBJETO ROBOT MANZANA BIKINI Z7
     20: ENTREGAR-OBJETO ROBOT MANZANA Z7 BRUJA
     21: MOVER-ROBOT ROBOT Z7 Z12 PIEDRA BIKINI
     22: MOVER-ROBOT ROBOT Z12 Z13 PIEDRA ROSA
     23: COGER-OBJETO ROBOT ROSA BIKINI Z13
```

Figura 5.1: Primera parte del plan encontrado para el quinto ejercicio



```
Archivo Editar Ver Bookmarks Preferencias Ayuda
22: MOVER-ROBOT ROBOT Z12 Z13 PIEDRA ROSA
23: COGER-OBJETO ROBOT ROSA BIKINI Z13
24: ENTREGAR-OBJETO ROBOT ROSA Z13 PRINCESA
25: MOVER-ROBOT ROBOT Z13 Z8 PIEDRA ROSA
26: MOVER-ROBOT ROBOT Z8 Z3 PIEDRA BIKINI
27: MOVER-ROBOT ROBOT Z3 Z2 PIEDRA BIKINI
28: COGER-OBJETO ROBOT ORO BIKINI Z2
29: MOVER-ROBOT ROBOT Z2 Z3 PIEDRA BIKINI
30: MOVER-ROBOT ROBOT Z3 Z4 PIEDRA BIKINI
31: DEJAR-OBJETO ROBOT ORO Z4
32: COGER-OBJETO ROBOT ORO BIKINI Z4
33: ENTREGAR-OBJETO ROBOT ORO Z4 PRINCIPE
34: MOVER-ROBOT ROBOT Z4 Z9 PIEDRA BIKINI
35: MOVER-ROBOT ROBOT Z9 Z10 PIEDRA BIKINI
36: COGER-OBJETO ROBOT OSCAR BIKINI Z10
37: MOVER-ROBOT ROBOT Z10 Z9 PIEDRA BIKINI
38: MOVER-ROBOT ROBOT Z9 Z8 PIEDRA BIKINI
39: MOVER-ROBOT ROBOT Z8 Z7 PIEDRA BIKINI
40: MOVER-ROBOT ROBOT Z7 Z6 PIEDRA BIKINI
41: DEJAR-OBJETO ROBOT OSCAR Z6
42: COGER-OBJETO ROBOT OSCAR BIKINI Z6
43: ENTREGAR-OBJETO ROBOT OSCAR Z6 LEONARDO
44: MOVER-ROBOT ROBOT Z6 Z7 PIEDRA BIKINI
45: SACAR-MOCHILA ROBOT ROSA
46: DEJAR-OBJETO ROBOT ROSA Z7
47: COGER-OBJETO ROBOT ROSA BIKINI Z7
```

Figura 5.2: Segunda parte del plan encontrado para el quinto ejercicio

```
Archivo  Editar  Ver  Bookmarks  Preferencias  Ayuda
44: MOVER-ROBOT ROBOT Z6 Z7 PIEDRA BIKINI
45: SACAR-MOCHILA ROBOT ROSA
46: DEJAR-OBJETO ROBOT ROSA Z7
47: COGER-OBJETO ROBOT ROSA BIKINI Z7
48: ENTREGAR-OBJETO ROBOT ROSA Z7 BRUJA
49: MOVER-ROBOT ROBOT Z7 Z2 PIEDRA BIKINI
50: COGER-OBJETO ROBOT ORO BIKINI Z2
51: MOVER-ROBOT ROBOT Z2 Z1 PIEDRA BIKINI
52: MOVER-ROBOT ROBOT Z1 Z6 PIEDRA BIKINI
53: DEJAR-OBJETO ROBOT ORO Z6
54: COGER-OBJETO ROBOT ORO BIKINI Z6
55: ENTREGAR-OBJETO ROBOT ORO Z6 LEONARDO

I
time spent:  0.02 seconds instantiating 1449 easy, 2368 hard action templates
             0.00 seconds reachability analysis, yielding 310 facts and 3807 actions
             0.00 seconds creating final representation with 303 relevant facts, 5 relevant
t fluents
             0.01 seconds computing LNF
             0.01 seconds building connectivity graph
             16.98 seconds searching, evaluating 43401 states, to a max depth of 12
             17.02 seconds total time

-----
/media/datos/Dropbox/3year/Segundo Cuatrimestre/TSI/Planificación/Metric-FF/Metric-FF(branch
:master*) »
```

Figura 5.3: Tercera parte del plan encontrado para el quinto ejercicio

Se puede observar por tanto que encuentra un plan con menos pasos, ya que en este caso son 55 pasos, y con el problema del ejercicio 4 eran 59 pasos.