

PRÁCTICA PDDL – TSI

JULIO FRESNEDA – JULIOFRESNEDAG@CORREO.UGR.ES – 49215154

EJERCICIO 1

a) Se han representado 4 tipos de objetos: Personaje, objeto, zona y jugador. Los tipos personaje y objeto tienen distintos subtipos:

Personaje tiene como subtipos: Princesa, príncipe, bruja, profesor, dicaprio.

Objeto tiene como subtipos: Manzana, rosa, algoritmo, oro, oscar

Se han organizado estos subtipos de esta forma por que es más cómodo de programar y más fácil de entender.

b) Los predicados son los siguientes:

```
(:predicates
  ( connected ?zona1 ?zona2 - zona )
  ( personaje_en ?personaje1 - personaje ?zona1 - zona )
  ( objeto_en ?objeto1 - objeto ?zona1 - zona )
  ( jugador_en ?j - jugador ?zona1 - zona )
  ( mano_vacia )
  ( mano_ocupada ?obj - objeto )
  ( personaje_tiene_objeto ?personaje1 - personaje )

  ( jugador_orientado_n ?j - jugador )
  ( jugador_orientado_s ?j - jugador )
  ( jugador_orientado_e ?j - jugador )
  ( jugador_orientado_o ?j - jugador )

  ( zona_orientada_n ?z1 - zona ?z2 - zona )
  ( zona_orientada_s ?z1 - zona ?z2 - zona )
  ( zona_orientada_e ?z1 - zona ?z2 - zona )
  ( zona_orientada_o ?z1 - zona ?z2 - zona )
```

Para conectar unas zonas con otras, se ha usado el predicado (connected ?zona1 ?zona2 – zona). Este predicado conecta las zonas que están contiguas.

Para describir la orientación de una zona respecto a la otra, se han usado los cuatro últimos predicados:

```
( zona_orientada_n ?z1 - zona ?z2 - zona )  
( zona_orientada_s ?z1 - zona ?z2 - zona )  
( zona_orientada_e ?z1 - zona ?z2 - zona )  
( zona_orientada_o ?z1 - zona ?z2 - zona )
```

Por ejemplo, si queremos decir que la zona20 está al norte de la zona19, usaríamos el predicado (zona_orientada_n zona20 zona19).

Para describir dónde están los personajes, objetos y el jugador se han usado estos predicados:

```
( personaje_en ?personaje1 - personaje ?zona1 - zona )  
( objeto_en ?objeto1 - objeto ?zona1 - zona )  
( jugador_en ?j - jugador ?zona1 - zona )
```

De esta forma cada personaje, objeto y el jugador están asociados a una zona.

Para describir si el jugador tiene algo o no en la mano, se han usado estos predicados:

```
( mano_vacia )  
( mano_ocupada ?obj - objeto )
```

Si un predicado se cumple, el otro no debe cumplirse.

Para indicar que un personaje tiene algún objeto (da igual cual sea) se ha usado este predicado:

```
( personaje_tiene_objeto ?personaje1 - personaje )
```

Como da igual qué objeto tenga el personaje, sólo usa personaje como argumento.

Para indicar la orientación del jugador, se ha usado estos predicados:

```
( jugador_orientado_n ?j - jugador )  
( jugador_orientado_s ?j - jugador )  
( jugador_orientado_e ?j - jugador )  
( jugador_orientado_o ?j - jugador )
```

El primer predicado indica que el jugador está orientado hacia el norte; el segundo, hacia el sur; etc. Cuando uno de estos predicados se cumple, los demás no deben cumplirse.

c) Las acciones son las siguientes:

Coger, dejar y entregar objeto:

```
(:action coger-objeto  
  :parameters ( ?obj - objeto ?z - zona ?j - jugador )  
  :precondition ( and (mano_vacia) (objeto_en ?obj ?z) (jugador_en ?j ?z) )  
  :effect ( and (not(mano_vacia)) (not(objeto_en ?obj ?z)) (mano_ocupada ?obj) )  
)  
  
(:action dejar-objeto  
  :parameters ( ?obj - objeto ?z - zona ?j - jugador )  
  :precondition ( and (mano_ocupada ?obj) (jugador_en ?j ?z) )  
  :effect ( and (not(mano_ocupada ?obj)) (mano_vacia) (objeto_en ?obj ?z) )  
)  
  
(:action entregar-objeto  
  :parameters ( ?obj - objeto ?z - zona ?p - personaje ?j - jugador )  
  :precondition ( and (personaje_en ?p ?z) (jugador_en ?j ?z) (mano_ocupada ?obj) )  
  :effect ( and (mano_vacia) (not(mano_ocupada ?obj)) (personaje_tiene_objeto ?p ) )  
)
```

En coger objeto, el jugador coge un objeto del suelo y lo lleva en la mano. Por lo tanto, las precondiciones son que el jugador deba tener la mano vacía y que el objeto esté en la misma zona que el jugador. Los efectos son que el jugador deja de tener la mano vacía y que el objeto ya no está en la zona.

En dejar objeto es justo al contrario, el jugador deja un objeto de su mano en el suelo. Por lo tanto, las precondiciones son que el jugador deba tener algún objeto en la mano, y los efectos son que la mano deja de estar ocupada, y el objeto ahora está en la zona donde está el jugador.

En entregar objeto, el jugador le entrega un objeto a un personaje. Las precondiciones para que esto ocurra son que el personaje y el jugador estén en la misma zona, y que el jugador tenga la

mano ocupada con el objeto. Los efectos son que el jugador tiene la mano vacía, y que el personaje tiene un objeto.

Girar derecha y girar izquierda:

```
(:action girar-derecha
  :parameters ( ?j - jugador )

  :effect ( and

    ( when (and( jugador_orientado_n ?j ))
      (and(not(jugador_orientado_n ?j )) (jugador_orientado_e ?j ) )
    )
    ( when (and( jugador_orientado_e ?j ))
      (and(not(jugador_orientado_e ?j )) (jugador_orientado_s ?j ) )
    )
    ( when (and( jugador_orientado_s ?j ))
      (and(not(jugador_orientado_s ?j )) (jugador_orientado_o ?j ) )
    )
    ( when (and( jugador_orientado_o ?j ))
      (and(not(jugador_orientado_o ?j )) (jugador_orientado_n ?j ) )
    )
  )
)

(:action girar-izquierda
  :parameters ( ?j - jugador )

  :effect ( and

    ( when (and( jugador_orientado_n ?j ))
      (and(not(jugador_orientado_n ?j )) (jugador_orientado_o ?j ) )
    )
    ( when (and( jugador_orientado_o ?j ))
      (and(not(jugador_orientado_o ?j )) (jugador_orientado_s ?j ) )
    )
    ( when (and( jugador_orientado_s ?j ))
      (and(not(jugador_orientado_s ?j )) (jugador_orientado_e ?j ) )
    )
    ( when (and( jugador_orientado_e ?j ))
      (and(not(jugador_orientado_e ?j )) (jugador_orientado_n ?j ) )
    )
  )
)
```

En estas acciones no se necesitan precondiciones. Lo que hacemos es decirle al jugador que cambie su orientación. Si gira a la derecha, el jugador pasa a orientarse a la derecha, ídem con la izquierda.

Por ejemplo, si el jugador está orientado hacia el sur y le decimos que gire a la derecha, pasará a estar orientado hacia el oeste.

Ir:

```
(:action ir
  :parameters ( ?origen - zona ?destino - zona ?j - jugador )
  :precondition(
    and
      (jugador_en ?j ?origen)
      ( or (connected ?destino ?origen) (connected ?origen ?destino) )
      ( or (and ( jugador_orientado_n ?j ) ( zona_orientada_n ?destino ?origen ) )
        (and ( jugador_orientado_e ?j ) ( zona_orientada_e ?destino ?origen ) )
        (and ( jugador_orientado_s ?j ) ( zona_orientada_s ?destino ?origen ) )
        (and ( jugador_orientado_o ?j ) ( zona_orientada_o ?destino ?origen ) )
      )
    )
  :effect( and (not( jugador_en ?j ?origen )) (jugador_en ?j ?destino) )
)
```

En esta acción el jugador se mueve desde una zona a otra que esté contigua, siempre y cuando esté bien orientado.

En esta acción, le pasamos como parámetros las zonas de origen y destino, y el jugador.

Las precondiciones son que el jugador esté en el origen, que destino y origen estén conectadas, y que el jugador esté orientado en la misma dirección donde está el destino. Por ejemplo, si el destino está al este del origen, el jugador debe estar mirando hacia el este.

Los efectos son que el jugador pasa de estar en el origen a estar en el destino.

c) Vamos a plantear un problema para ver cómo funciona nuestro modelo.

Para ello, nuestro mapa tendrá 25 zonas, con 5 personajes distintos y un mínimo de 5 objetos. El objetivo de este problema será que cada personaje tenga al menos un objeto.

El mapa sería el siguiente:

1	2 Profesor	3 Manzana	4	5
6	7	8	9	10 Oro
11	12	13 Príncipe Algoritmo	14	15
16 Bruja	17	18	19 Óscar	20
21 Rosa	22	23 Princesa	24	25 DiCaprio

Antes de ver cómo se comporta el plan con el objetivo propuesto, vamos a hacer una pequeña prueba con un objetivo mucho más simple: Que la princesa tenga objeto. Para ello, cargaremos el fichero Ej1problema2.pddl.

El resultado es el siguiente:

```
ff: found legal plan as follows

step    0: GIRAR-IZQUIERDA JUGADOR1
         1: GIRAR-IZQUIERDA JUGADOR1
         2: IR Z1 Z6 JUGADOR1
         3: IR Z6 Z11 JUGADOR1
         4: IR Z11 Z16 JUGADOR1
         5: IR Z16 Z21 JUGADOR1
         6: GIRAR-IZQUIERDA JUGADOR1
         7: COGER-OBJETO ROSA1 Z21 JUGADOR1
         8: IR Z21 Z22 JUGADOR1
         9: IR Z22 Z23 JUGADOR1
        10: ENTREGAR-OBJETO ROSA1 Z23 PRINCESA1 JUGADOR1
```

Como vemos, funciona bien.

Vamos a buscar el plan propuesto por el ejercicio a ver cómo se comporta. Recordemos que nuestro objetivo es el siguiente:

```
(:goal
  ( and ( personaje_tiene_objeto princesa1 ) ( personaje_tiene_objeto principel )
        ( personaje_tiene_objeto bruja1 ) ( personaje_tiene_objeto profesor1 )
        ( personaje_tiene_objeto dicaprio1 ) )
)
```

El plan encontrado es el siguiente.

und legal plan as follows

```
0: GIRAR-IZQUIERDA JUGADOR1
1: GIRAR-IZQUIERDA JUGADOR1
2: IR Z1 Z6 JUGADOR1
3: IR Z6 Z11 JUGADOR1
4: IR Z11 Z16 JUGADOR1
5: IR Z16 Z21 JUGADOR1
6: GIRAR-IZQUIERDA JUGADOR1
7: COGER-OBJETO ROSA1 Z21 JUGADOR1
8: IR Z21 Z22 JUGADOR1
9: IR Z22 Z23 JUGADOR1
10: IR Z23 Z24 JUGADOR1
11: IR Z24 Z25 JUGADOR1
12: GIRAR-IZQUIERDA JUGADOR1
13: GIRAR-IZQUIERDA JUGADOR1
14: ENTREGAR-OBJETO ROSA1 Z25 DICAPRIO1 JUGADOR1
15: IR Z25 Z24 JUGADOR1
16: IR Z24 Z23 JUGADOR1
17: IR Z23 Z22 JUGADOR1
18: IR Z22 Z21 JUGADOR1
19: GIRAR-IZQUIERDA JUGADOR1
20: GIRAR-IZQUIERDA JUGADOR1
21: GIRAR-IZQUIERDA JUGADOR1
22: IR Z21 Z16 JUGADOR1
23: IR Z16 Z11 JUGADOR1
24: GIRAR-DERECHA JUGADOR1
25: IR Z11 Z12 JUGADOR1
26: IR Z12 Z13 JUGADOR1
27: GIRAR-DERECHA JUGADOR1
28: GIRAR-DERECHA JUGADOR1
29: COGER-OBJETO ALGORITMO1 Z13 JUGADOR1
30: ENTREGAR-OBJETO ALGORITMO1 Z13 PRINCIPE1 JUGADOR1
31: IR Z13 Z12 JUGADOR1
32: IR Z12 Z11 JUGADOR1
33: GIRAR-IZQUIERDA JUGADOR1
34: IR Z11 Z16 JUGADOR1
35: IR Z16 Z21 JUGADOR1
36: GIRAR-DERECHA JUGADOR1
37: GIRAR-DERECHA JUGADOR1
```

```
37: GIRAR-DERECHA JUGADOR1
38: IR Z21 Z16 JUGADOR1
39: IR Z16 Z11 JUGADOR1
40: IR Z11 Z6 JUGADOR1
41: IR Z6 Z1 JUGADOR1
42: GIRAR-DERECHA JUGADOR1
43: IR Z1 Z2 JUGADOR1
44: IR Z2 Z3 JUGADOR1
45: GIRAR-IZQUIERDA JUGADOR1
46: GIRAR-IZQUIERDA JUGADOR1
47: COGER-OBJETO MANZANA1 Z3 JUGADOR1
48: IR Z3 Z2 JUGADOR1
49: IR Z2 Z1 JUGADOR1
50: GIRAR-IZQUIERDA JUGADOR1
51: IR Z1 Z6 JUGADOR1
52: IR Z6 Z11 JUGADOR1
53: IR Z11 Z16 JUGADOR1
54: IR Z16 Z21 JUGADOR1
55: GIRAR-IZQUIERDA JUGADOR1
56: IR Z21 Z22 JUGADOR1
57: IR Z22 Z23 JUGADOR1
58: ENTREGAR-OBJETO MANZANA1 Z23 PRINCESA1 JUGADOR1
59: IR Z23 Z24 JUGADOR1
60: IR Z24 Z25 JUGADOR1
61: GIRAR-IZQUIERDA JUGADOR1
62: IR Z25 Z20 JUGADOR1
63: IR Z20 Z15 JUGADOR1
64: IR Z15 Z10 JUGADOR1
65: GIRAR-IZQUIERDA JUGADOR1
66: GIRAR-IZQUIERDA JUGADOR1
67: COGER-OBJETO OR01 Z10 JUGADOR1
68: IR Z10 Z15 JUGADOR1
69: IR Z15 Z20 JUGADOR1
70: IR Z20 Z25 JUGADOR1
71: GIRAR-DERECHA JUGADOR1
72: IR Z25 Z24 JUGADOR1
73: IR Z24 Z23 JUGADOR1
74: IR Z23 Z22 JUGADOR1
75: IR Z22 Z21 JUGADOR1
76: GIRAR-IZQUIERDA JUGADOR1
77: GIRAR-IZQUIERDA JUGADOR1
78: GIRAR-IZQUIERDA JUGADOR1
```

```
79: IR Z21 Z16 JUGADOR1
80: GIRAR-DERECHA JUGADOR1
81: ENTREGAR-OBJETO OR01 Z16 BRUJA1 JUGADOR1
82: GIRAR-DERECHA JUGADOR1
83: IR Z16 Z21 JUGADOR1
84: GIRAR-IZQUIERDA JUGADOR1
85: IR Z21 Z22 JUGADOR1
86: IR Z22 Z23 JUGADOR1
87: IR Z23 Z24 JUGADOR1
88: IR Z24 Z25 JUGADOR1
89: GIRAR-IZQUIERDA JUGADOR1
90: IR Z25 Z20 JUGADOR1
91: GIRAR-IZQUIERDA JUGADOR1
92: IR Z20 Z19 JUGADOR1
93: GIRAR-IZQUIERDA JUGADOR1
94: GIRAR-IZQUIERDA JUGADOR1
95: COGER-OBJETO OSCAR1 Z19 JUGADOR1
96: IR Z19 Z20 JUGADOR1
97: GIRAR-DERECHA JUGADOR1
98: IR Z20 Z25 JUGADOR1
99: GIRAR-DERECHA JUGADOR1
100: IR Z25 Z24 JUGADOR1
101: IR Z24 Z23 JUGADOR1
102: IR Z23 Z22 JUGADOR1
103: IR Z22 Z21 JUGADOR1
104: GIRAR-DERECHA JUGADOR1
105: IR Z21 Z16 JUGADOR1
106: IR Z16 Z11 JUGADOR1
107: IR Z11 Z6 JUGADOR1
108: IR Z6 Z1 JUGADOR1
109: GIRAR-DERECHA JUGADOR1
110: IR Z1 Z2 JUGADOR1
111: ENTREGAR-OBJETO OSCAR1 Z2 PROFESOR1 JUGADOR1
```


Vemos que es un plan bastante largo, pero si lo seguimos poco a poco podemos comprobar que se cumple, y cada personaje tiene su objeto.

EJERCICIO 2.

En este nuevo dominio, vamos a asignarle una longitud a cada camino entre zonas contiguas.

a) Las modificaciones del dominio son las siguientes.

Ahora, usaremos dos nuevas funciones:

```
(:functions
  ( distancia ?z1 ?z2 - zona )
  ( distancia_total )
)
```

La primera función asigna la distancia que hay entre dos zonas contiguas, y la segunda función es el contador de la distancia total recorrida.

En la acción de ir de una zona a otra, se ha añadido un nuevo efecto: Cuando se desplaza, a la distancia total se le suma la distancia recorrida.

```
(:action ir
  :parameters ( ?origen - zona ?destino - zona ?j - jugador )
  :precondition(
    and
      (jugador_en ?j ?origen)
      ( or (connected ?destino ?origen) (connected ?origen ?destino) )
      ( or (and ( jugador_orientado_n ?j ) ( zona_orientada_n ?destino ?origen ) )
        (and ( jugador_orientado_e ?j ) ( zona_orientada_e ?destino ?origen ) )
        (and ( jugador_orientado_s ?j ) ( zona_orientada_s ?destino ?origen ) )
        (and ( jugador_orientado_o ?j ) ( zona_orientada_o ?destino ?origen ) )
      )
    )
  :effect( and (not( jugador_en ?j ?origen )) (jugador_en ?j ?destino)
    ( increase (distancia_total) (distancia ?origen ?destino ) ) )
)
```

Para comprobar que funciona, vamos a plantear lo siguiente.

En los problemas que plantearemos, intentaremos reducir al máximo la distancia recorrida:

```
(:metric minimize (distancia_total))
```

Para ver que funciona, vamos a plantear dos problemas: En ambos se le deberá entregar un objeto al profesor, pero las distancias de cada problema serán diferentes.

En el primer problema, cada zona está separada de la otra por 1 unidad de distancia. En el plan obtenido vemos que el jugador obtiene la manzana que está a la derecha (Z3) del profesor (Z2):

```
step    0: GIRAR-DERECHA JUGADOR1
        1: IR Z1 Z2 JUGADOR1
        2: IR Z2 Z3 JUGADOR1
        3: GIRAR-IZQUIERDA JUGADOR1
        4: GIRAR-IZQUIERDA JUGADOR1
        5: COGER-OBJETO MANZANA1 Z3 JUGADOR1
        6: IR Z3 Z2 JUGADOR1
        7: ENTREGAR-OBJETO MANZANA1 Z2 PROFESOR1 JUGADOR1
```

En el segundo problema, vamos a asignar a la distancia entre Z3 y Z2, 100 unidades de longitud. Veamos el plan que obtenemos. Recordemos que hay que ejecutar ./ff con las opciones para que encuentren la solución óptima:-O-h 1-g 1.

El plan encontrado es el siguiente:

```
step    0: GIRAR-DERECHA JUGADOR1
        1: GIRAR-DERECHA JUGADOR1
        2: IR Z1 Z6 JUGADOR1
        3: IR Z6 Z11 JUGADOR1
        4: IR Z11 Z16 JUGADOR1
        5: IR Z16 Z21 JUGADOR1
        6: COGER-OBJETO ROSA1 Z21 JUGADOR1
        7: GIRAR-DERECHA JUGADOR1
        8: GIRAR-DERECHA JUGADOR1
        9: IR Z21 Z16 JUGADOR1
       10: IR Z16 Z11 JUGADOR1
       11: IR Z11 Z6 JUGADOR1
       12: IR Z6 Z1 JUGADOR1
       13: GIRAR-DERECHA JUGADOR1
       14: IR Z1 Z2 JUGADOR1
       15: DEJAR-OBJETO ROSA1 Z2 JUGADOR1
       16: GIRAR-DERECHA JUGADOR1
       17: GIRAR-DERECHA JUGADOR1
       18: COGER-OBJETO ROSA1 Z2 JUGADOR1
       19: ENTREGAR-OBJETO ROSA1 Z2 PROFESOR1 JUGADOR1
```

Vemos que se ha olvidado de ir a por la manzana, ya que el coste era muy alto, y ha ido a por la rosa, que es el objeto que más cerca está del profesor.

EJERCICIO 3.

En este problema ahora hay cinco tipos distintos de zonas: Arena, agua, piedra, bosque y precipicio. Además, hay dos nuevos objetos: Zapatillas y bikini.

El jugador ahora tiene una mochila, donde podrá guardar un objeto.

En este problema y en los siguientes, para ahorrar tiempo de ejecución, he desactivado la búsqueda óptima en cuanto a distancias.

Se han realizado las siguientes modificaciones:

En el dominio, se han añadido los siguientes predicados:

```
( zona_agua ?z - zona )
( zona_precipicio ?z - zona )
( zona_bosque ?z - zona )
( zona_piedra ?z - zona )
( zona_arena ?z - zona )

( en_mochila ?obj - objeto )

( es_bikini ?obj - objeto )
( es_zapas ?obj - objeto )
```

Los cinco primeros predicados clasifican la zona según el tipo que sean.

Se ha añadido también un predicado para decir si tenemos un objeto en la mochila, y dos predicados para decirnos si un objeto es un bikini o son zapatillas.

B) Ahora, además de las acciones de coger, dejar y entregar un objeto, están las acciones de meter y sacar un objeto en la mochila. Recordemos que para meter un objeto en la mochila, debemos tener ese objeto en la mano y la mochila debe estar vacía, y para sacar un objeto de la mochila, la mano debe de estar vacía.

Las nuevas acciones son las siguientes:

```
(:action meter-mochila
  :parameters ( ?obj - objeto )
  :precondition ( and (mano_ocupada ?obj)
    (forall( ?otro - objeto )(not(en_mochila ?otro))))
  :effect ( and (mano_vacia) (not(mano_ocupada ?obj)) ( en_mochila ?obj ) )
)

(:action sacar-mochila
  :parameters ( ?obj - objeto )
  :precondition ( and( mano_vacia) ( en_mochila ?obj ))
  :effect ( and ( not(mano_vacia)) (mano_ocupada ?obj ) (not(en_mochila ?obj)) )
)
```

Como vemos, para meter un objeto en la mochila, tenemos como precondition que debemos tener ese objeto en la mano, y que la mochila debe estar vacía. Como efecto vemos que dejamos la mano vacía y la mochila tiene objeto.

Para sacar un objeto de la mochila, es justo al contrario, como preconditiones debemos tener la mano vacía y el objeto en la mochila, y como efecto tenemos el objeto en la mano y la mochila la dejamos vacía.

A) En la acción de “Ir” también hay diferencias:

```
(:action ir
  :parameters ( ?origen - zona ?destino - zona ?j - jugador ?obj - objeto )
  :precondition(
    and
      (jugador_en ?j ?origen)
      ( or (connected ?destino ?origen) (connected ?origen ?destino) )
      ( or (and ( jugador_orientado_n ?j ) ( zona_orientada_n ?destino ?origen ) )
        ( and ( jugador_orientado_e ?j ) ( zona_orientada_e ?destino ?origen ) )
        ( and ( jugador_orientado_s ?j ) ( zona_orientada_s ?destino ?origen ) )
        ( and ( jugador_orientado_o ?j ) ( zona_orientada_o ?destino ?origen ) )
      )
      ( or ( zona_arena ?destino ) ( zona_piedra ?destino )
        ( and ( zona_agua ?destino ) ( es_bikini ?obj ) ( or ( en_mochila ?obj ) (mano_ocupada ?obj) ) )
        ( and ( zona_bosque ?destino ) (es_zapas ?obj) ( or ( en_mochila ?obj ) (mano_ocupada ?obj) ) )
      )
    )
  :effect( and (not( jugador_en ?j ?origen )) (jugador_en ?j ?destino)
    ( increase (distancia_total) (distancia ?origen ?destino) ) )
)
```

Como vemos en las precondiciones, si el destino es agua, debemos tener un bikini en la mano o en la mochila, y si el destino es bosque, debemos tener unas zapatillas.

En la acción de dejar objeto también hay una diferencia:

```
(:action dejar-objeto
  :parameters ( ?obj - objeto ?z - zona ?j - jugador )
  :precondition ( and (mano_ocupada ?obj) (jugador_en ?j ?z)
    ( or (not(zona_bosque ?z )) (and (zona_bosque ?z ) (not(es_zapas ?obj)) ) )
    ( or (not(zona_agua ?z )) (and (zona_agua ?z ) (not(es_bikini ?obj) ) ) )
  )
  :effect ( and (not(mano_ocupada ?obj)) (mano_vacia) (objeto_en ?obj ?z) )
)
```

Ahora no se puede dejar un objeto en el bosque si tenemos las zapatillas en la mano, es decir, no podemos dejar las zapatillas por que las estamos usando. Igualmente con el agua y el bikini.

C) El nuevo mapa, con las zonas añadidas, es el siguiente:

1 Piedra	2 Piedra Profesor	3 Agua Manzana	4 Agua Zapas	5 Agua
6 Piedra	7 Piedra Zapas	8 Precipicio	9 Agua	10 Agua Oro
11 Piedra	12 Piedra	13 Agua Príncipe Algoritmo	14 Agua	15 Arena
16 Arena Bruja	17 Arena	18 Arena Bikini	19 Arena Óscar	20 Bosque
21 Arena Rosa	22 Bosque	23 Bosque Princesa	24 Bosque Bikini	25 Bosque DiCaprio

Vamos a ver cómo funciona, con un ejemplo sencillo: Vamos a darle un objeto al príncipe. Pero hay un problema, el príncipe está sobre agua. Vamos a ver cómo se resuelve el problema:

PD: En las acciones de "IR", siempre a la derecha aparece algún objeto, como zapas2 en este caso. Esto es por que la acción "IR" necesita un objeto como argumento, pero se puede ignorar en la lectura del plan.

```
0: GIRAR-IZQUIERDA JUGADOR1
1: GIRAR-IZQUIERDA JUGADOR1
2: IR Z1 Z6 JUGADOR1 ZAPAS2
3: GIRAR-IZQUIERDA JUGADOR1
4: IR Z6 Z7 JUGADOR1 ZAPAS2
5: GIRAR-IZQUIERDA JUGADOR1
6: GIRAR-IZQUIERDA JUGADOR1
7: COGER-OBJETO ZAPAS1 Z7 JUGADOR1
8: IR Z7 Z6 JUGADOR1 ZAPAS2
9: GIRAR-IZQUIERDA JUGADOR1
10: IR Z6 Z11 JUGADOR1 ZAPAS2
11: METER-MOCHILA ZAPAS1
12: IR Z11 Z16 JUGADOR1 ZAPAS2
13: IR Z16 Z21 JUGADOR1 ZAPAS2
14: GIRAR-IZQUIERDA JUGADOR1
15: IR Z21 Z22 JUGADOR1 ZAPAS1
16: IR Z22 Z23 JUGADOR1 ZAPAS1
17: IR Z23 Z24 JUGADOR1 ZAPAS1
18: GIRAR-IZQUIERDA JUGADOR1
19: GIRAR-IZQUIERDA JUGADOR1
20: COGER-OBJETO BIKINI2 Z24 JUGADOR1
21: IR Z24 Z23 JUGADOR1 ZAPAS1
22: IR Z23 Z22 JUGADOR1 ZAPAS1
23: IR Z22 Z21 JUGADOR1 ZAPAS2
24: GIRAR-IZQUIERDA JUGADOR1
25: GIRAR-IZQUIERDA JUGADOR1
26: GIRAR-IZQUIERDA JUGADOR1
27: IR Z21 Z16 JUGADOR1 ZAPAS2
28: IR Z16 Z11 JUGADOR1 ZAPAS2
29: GIRAR-DERECHA JUGADOR1
30: IR Z11 Z12 JUGADOR1 ZAPAS2
31: DEJAR-OBJETO BIKINI2 Z12 JUGADOR1
32: SACAR-MOCHILA ZAPAS1
33: DEJAR-OBJETO ZAPAS1 Z12 JUGADOR1
34: COGER-OBJETO BIKINI2 Z12 JUGADOR1
35: IR Z12 Z13 JUGADOR1 BIKINI2
36: METER-MOCHILA BIKINI2
37: COGER-OBJETO ALGORITMO1 Z13 JUGADOR1
38: ENTREGAR-OBJETO ALGORITMO1 Z13 PRINCIPE1 JUGADOR1
```

Vemos que para pasar por el bosque coge las zapatillas, y las mete en la mochila. Ya en el bosque, coge el bikini. Para coger el algoritmo y dárselo al príncipe, necesita dejar un objeto, pues tiene las zapatillas en la mochila y el bikini en la mano. Por lo tanto, deja el bikini en el suelo, saca las zapatillas de la mochila, deja las zapatillas en el suelo, coge el bikini del suelo, se lo mete en la mochila y procede a entrar al agua y darle el algoritmo al príncipe.

También tenemos un problema2, donde se busca un plan para el objetivo del ejercicio 1: Que cada personaje tenga al menos un objeto. Veamos el plan:

```
0: GIRAR-IZQUIERDA JUGADOR1
1: GIRAR-IZQUIERDA JUGADOR1
2: IR Z1 Z6 JUGADOR1 ZAPAS2
3: GIRAR-IZQUIERDA JUGADOR1
4: IR Z6 Z7 JUGADOR1 ZAPAS2
5: GIRAR-IZQUIERDA JUGADOR1
6: GIRAR-IZQUIERDA JUGADOR1
7: COGER-OBJETO ZAPAS1 Z7 JUGADOR1
8: IR Z7 Z6 JUGADOR1 ZAPAS2
9: GIRAR-IZQUIERDA JUGADOR1
10: METER-MOCHILA ZAPAS1
11: IR Z6 Z11 JUGADOR1 ZAPAS2
12: IR Z11 Z16 JUGADOR1 ZAPAS2
13: IR Z16 Z21 JUGADOR1 ZAPAS2
14: GIRAR-IZQUIERDA JUGADOR1
15: GIRAR-IZQUIERDA JUGADOR1
16: COGER-OBJETO ROSA1 Z21 JUGADOR1
17: IR Z21 Z16 JUGADOR1 ZAPAS2
18: GIRAR-IZQUIERDA JUGADOR1
19: GIRAR-IZQUIERDA JUGADOR1
20: ENTREGAR-OBJETO ROSA1 Z16 BRUJA1 JUGADOR1
21: IR Z16 Z21 JUGADOR1 ZAPAS2
22: GIRAR-IZQUIERDA JUGADOR1
23: IR Z21 Z22 JUGADOR1 ZAPAS1
24: IR Z22 Z23 JUGADOR1 ZAPAS1
25: IR Z23 Z24 JUGADOR1 ZAPAS1
26: IR Z24 Z25 JUGADOR1 ZAPAS1
27: GIRAR-IZQUIERDA JUGADOR1
28: IR Z25 Z20 JUGADOR1 ZAPAS1
29: GIRAR-IZQUIERDA JUGADOR1
30: IR Z20 Z19 JUGADOR1 ZAPAS2
31: GIRAR-IZQUIERDA JUGADOR1
32: GIRAR-IZQUIERDA JUGADOR1
33: COGER-OBJETO OSCAR1 Z19 JUGADOR1
34: IR Z19 Z20 JUGADOR1 ZAPAS1
35: GIRAR-DERECHA JUGADOR1
36: IR Z20 Z25 JUGADOR1 ZAPAS1
37: ENTREGAR-OBJETO OSCAR1 Z25 DICAPRIO1 JUGADOR1
38: GIRAR-DERECHA JUGADOR1
39: IR Z25 Z24 JUGADOR1 ZAPAS1
40: COGER-OBJETO BIKINI2 Z24 JUGADOR1
41: IR Z24 Z23 JUGADOR1 ZAPAS1
42: IR Z23 Z22 JUGADOR1 ZAPAS1
43: IR Z22 Z21 JUGADOR1 ZAPAS2
44: GIRAR-DERECHA JUGADOR1
45: DEJAR-OBJETO BIKINI2 Z21 JUGADOR1
46: SACAR-MOCHILA ZAPAS1
47: DEJAR-OBJETO ZAPAS1 Z21 JUGADOR1
48: COGER-OBJETO BIKINI2 Z21 JUGADOR1
49: IR Z21 Z16 JUGADOR1 ZAPAS2
```

```
50: METER-MOCHILA BIKINI2
51: IR Z16 Z11 JUGADOR1 ZAPAS2
52: GIRAR-DERECHA JUGADOR1
53: IR Z11 Z12 JUGADOR1 ZAPAS2
54: IR Z12 Z13 JUGADOR1 BIKINI2
55: GIRAR-IZQUIERDA JUGADOR1
56: COGER-OBJETO ALGORITMO1 Z13 JUGADOR1
57: GIRAR-IZQUIERDA JUGADOR1
58: ENTREGAR-OBJETO ALGORITMO1 Z13 PRINCIPE1 JUGADOR1
59: IR Z13 Z12 JUGADOR1 ZAPAS2
60: IR Z12 Z11 JUGADOR1 ZAPAS2
61: GIRAR-IZQUIERDA JUGADOR1
62: IR Z11 Z16 JUGADOR1 ZAPAS2
63: IR Z16 Z21 JUGADOR1 ZAPAS2
64: GIRAR-IZQUIERDA JUGADOR1
65: GIRAR-IZQUIERDA JUGADOR1
66: IR Z21 Z16 JUGADOR1 ZAPAS2
67: IR Z16 Z11 JUGADOR1 ZAPAS2
68: IR Z11 Z6 JUGADOR1 ZAPAS2
69: IR Z6 Z1 JUGADOR1 ZAPAS2
70: GIRAR-DERECHA JUGADOR1
71: IR Z1 Z2 JUGADOR1 ZAPAS2
72: IR Z2 Z3 JUGADOR1 BIKINI2
73: GIRAR-IZQUIERDA JUGADOR1
74: GIRAR-IZQUIERDA JUGADOR1
75: COGER-OBJETO MANZANA1 Z3 JUGADOR1
76: IR Z3 Z2 JUGADOR1 ZAPAS2
77: ENTREGAR-OBJETO MANZANA1 Z2 PROFESOR1 JUGADOR1
78: IR Z2 Z1 JUGADOR1 ZAPAS2
79: GIRAR-IZQUIERDA JUGADOR1
80: IR Z1 Z6 JUGADOR1 ZAPAS2
81: IR Z6 Z11 JUGADOR1 ZAPAS2
82: IR Z11 Z16 JUGADOR1 ZAPAS2
83: IR Z16 Z21 JUGADOR1 ZAPAS2
84: GIRAR-IZQUIERDA JUGADOR1
85: COGER-OBJETO ZAPAS1 Z21 JUGADOR1
86: IR Z21 Z22 JUGADOR1 ZAPAS1
87: IR Z22 Z23 JUGADOR1 ZAPAS1
88: IR Z23 Z24 JUGADOR1 ZAPAS1
89: IR Z24 Z25 JUGADOR1 ZAPAS1
90: GIRAR-IZQUIERDA JUGADOR1
91: IR Z25 Z20 JUGADOR1 ZAPAS1
92: IR Z20 Z15 JUGADOR1 ZAPAS2
93: IR Z15 Z10 JUGADOR1 BIKINI2
94: GIRAR-IZQUIERDA JUGADOR1
95: GIRAR-IZQUIERDA JUGADOR1
96: IR Z10 Z15 JUGADOR1 ZAPAS2
97: GIRAR-IZQUIERDA JUGADOR1
98: GIRAR-IZQUIERDA JUGADOR1
```

```
99: DEJAR-OBJETO ZAPAS1 Z15 JUGADOR1
100: IR Z15 Z10 JUGADOR1 BIKINI2
101: GIRAR-IZQUIERDA JUGADOR1
102: GIRAR-IZQUIERDA JUGADOR1
103: COGER-OBJETO ORO1 Z10 JUGADOR1
104: IR Z10 Z15 JUGADOR1 ZAPAS2
105: DEJAR-OBJETO ORO1 Z15 JUGADOR1
106: SACAR-MOCHILA BIKINI2
107: DEJAR-OBJETO BIKINI2 Z15 JUGADOR1
108: COGER-OBJETO ZAPAS1 Z15 JUGADOR1
109: METER-MOCHILA ZAPAS1
110: COGER-OBJETO ORO1 Z15 JUGADOR1
111: IR Z15 Z20 JUGADOR1 ZAPAS1
112: IR Z20 Z25 JUGADOR1 ZAPAS1
113: GIRAR-DERECHA JUGADOR1
114: IR Z25 Z24 JUGADOR1 ZAPAS1
115: IR Z24 Z23 JUGADOR1 ZAPAS1
116: ENTREGAR-OBJETO ORO1 Z23 PRINCESA1 JUGADOR1
```

Vemos que es bastante largo, pero si lo seguimos poco a poco vemos que el plan es correcto.

EJERCICIO 4:

Ahora cada vez que nuestro jugador entrega objetos a un personaje, obtiene puntos. La tabla de puntos es la siguiente:

	Leonardo	Princesa	Bruja	Profesor	Príncipe
Oscar	10	5	4	3	1
Rosa	1	10	5	4	3
Manzana	3	1	10	5	4
Algoritmo	4	3	1	10	5
Oro	5	4	3	1	10

A) Para cumplir ésto se han añadido dos funciones de puntos:

```
(:functions
  ( distancia ?z1 ?z2 - zona )
  ( distancia_total )

  ( puntos ?obj - objeto ?p - personaje )
  ( puntos_totales )

)
```

En la primera función de puntos, se asignan las correspondencias de puntos objeto-personaje.

En la función de puntos_totales, llevamos la cuenta de los puntos obtenidos.

Nuestra acción de entregar-objeto, también ha cambiado:

```
(:action entregar-objeto
  :parameters ( ?obj - objeto ?z - zona ?p - personaje ?j - jugador )
  :precondition ( and (personaje_en ?p ?z) (jugador_en ?j ?z) (mano_ocupada ?obj) (not(es_bikini ?obj))
                    (not(es_zapas ?obj)) )
  :effect ( and (mano_vacia) (not(mano_ocupada ?obj)) (personaje_tiene_objeto ?p )
              ( increase ( puntos_totales) ( puntos ?obj ?p ) ) )
)
```

Vemos que en los efectos, le incrementamos a los puntos totales los puntos que hemos conseguido por darle ese objeto a ese personaje.

B) Vamos a probar ésto con un problema. Vamos a intentar obtener 20 puntos, dando igual si hay jugadores que se quedan sin objeto.

```
0: GIRAR-IZQUIERDA JUGADOR1
1: GIRAR-IZQUIERDA JUGADOR1
2: IR Z1 Z6 JUGADOR1 ZAPAS2
3: IR Z6 Z11 JUGADOR1 ZAPAS2
4: IR Z11 Z16 JUGADOR1 ZAPAS2
5: IR Z16 Z21 JUGADOR1 ZAPAS2
6: GIRAR-IZQUIERDA JUGADOR1
7: GIRAR-IZQUIERDA JUGADOR1
8: COGER-OBJETO ROSA1 Z21 JUGADOR1
9: IR Z21 Z16 JUGADOR1 ZAPAS2
10: ENTREGAR-OBJETO ROSA1 Z16 BRUJA1 JUGADOR1
11: IR Z16 Z11 JUGADOR1 ZAPAS2
12: IR Z11 Z6 JUGADOR1 ZAPAS2
13: GIRAR-DERECHA JUGADOR1
14: IR Z6 Z7 JUGADOR1 ZAPAS2
15: GIRAR-IZQUIERDA JUGADOR1
16: GIRAR-IZQUIERDA JUGADOR1
17: COGER-OBJETO ZAPAS1 Z7 JUGADOR1
18: IR Z7 Z6 JUGADOR1 ZAPAS2
19: GIRAR-IZQUIERDA JUGADOR1
20: IR Z6 Z11 JUGADOR1 ZAPAS2
21: METER-MOCHILA ZAPAS1
22: IR Z11 Z16 JUGADOR1 ZAPAS2
23: IR Z16 Z21 JUGADOR1 ZAPAS2
24: GIRAR-IZQUIERDA JUGADOR1
25: IR Z21 Z22 JUGADOR1 ZAPAS1
26: IR Z22 Z23 JUGADOR1 ZAPAS1
27: IR Z23 Z24 JUGADOR1 ZAPAS1
28: IR Z24 Z25 JUGADOR1 ZAPAS1
29: GIRAR-IZQUIERDA JUGADOR1
30: IR Z25 Z20 JUGADOR1 ZAPAS1
31: GIRAR-IZQUIERDA JUGADOR1
32: IR Z20 Z19 JUGADOR1 ZAPAS2
33: GIRAR-IZQUIERDA JUGADOR1
34: GIRAR-IZQUIERDA JUGADOR1
35: COGER-OBJETO OSCAR1 Z19 JUGADOR1
36: IR Z19 Z20 JUGADOR1 ZAPAS1
37: GIRAR-DERECHA JUGADOR1
38: IR Z20 Z25 JUGADOR1 ZAPAS1
39: GIRAR-DERECHA JUGADOR1
40: ENTREGAR-OBJETO OSCAR1 Z25 DICAPRIO1 JUGADOR1
41: IR Z25 Z24 JUGADOR1 ZAPAS1
42: COGER-OBJETO BIKINI2 Z24 JUGADOR1
43: IR Z24 Z23 JUGADOR1 ZAPAS1
44: IR Z23 Z22 JUGADOR1 ZAPAS1
45: IR Z22 Z21 JUGADOR1 ZAPAS2
46: GIRAR-DERECHA JUGADOR1
47: DEJAR-OBJETO BIKINI2 Z21 JUGADOR1
48: SACAR-MOCHILA ZAPAS1
49: DEJAR-OBJETO ZAPAS1 Z21 JUGADOR1
50: COGER-OBJETO BIKINI2 Z21 JUGADOR1
51: IR Z21 Z16 JUGADOR1 ZAPAS2
52: METER-MOCHILA BIKINI2
53: IR Z16 Z11 JUGADOR1 ZAPAS2
54: GIRAR-DERECHA JUGADOR1
55: IR Z11 Z12 JUGADOR1 ZAPAS2
56: IR Z12 Z13 JUGADOR1 BIKINI2
57: COGER-OBJETO ALGORITMO1 Z13 JUGADOR1
58: ENTREGAR-OBJETO ALGORITMO1 Z13 PRINCIPE1 JUGADOR1
```

Como vemos, el plan ha acabado cuando se han alcanzado los 20 puntos.

Vamos a probar con otro problema más interesante, el problema2, que consiste en alcanzar 20 puntos habiendo únicamente 2 rosas, por lo que se le deberán entregar las 2 a la princesa.

```
0: GIRAR-IZQUIERDA JUGADOR1
1: GIRAR-IZQUIERDA JUGADOR1
2: IR Z1 Z6 JUGADOR1 ZAPAS2
3: GIRAR-IZQUIERDA JUGADOR1
4: IR Z6 Z7 JUGADOR1 ZAPAS2
5: GIRAR-IZQUIERDA JUGADOR1
6: GIRAR-IZQUIERDA JUGADOR1
7: COGER-OBJETO ZAPAS1 Z7 JUGADOR1
8: IR Z7 Z6 JUGADOR1 ZAPAS2
9: GIRAR-IZQUIERDA JUGADOR1
10: METER-MOCHILA ZAPAS1
11: GIRAR-IZQUIERDA JUGADOR1
12: GIRAR-IZQUIERDA JUGADOR1
13: IR Z6 Z1 JUGADOR1 ZAPAS2
14: GIRAR-DERECHA JUGADOR1
15: GIRAR-DERECHA JUGADOR1
16: IR Z1 Z6 JUGADOR1 ZAPAS2
17: IR Z6 Z11 JUGADOR1 ZAPAS2
18: IR Z11 Z16 JUGADOR1 ZAPAS2
19: IR Z16 Z21 JUGADOR1 ZAPAS2
20: GIRAR-IZQUIERDA JUGADOR1
21: IR Z21 Z22 JUGADOR1 ZAPAS1
22: IR Z22 Z23 JUGADOR1 ZAPAS1
23: IR Z23 Z24 JUGADOR1 ZAPAS1
24: COGER-OBJETO BIKINI2 Z24 JUGADOR1
25: IR Z24 Z25 JUGADOR1 ZAPAS1
26: GIRAR-IZQUIERDA JUGADOR1
27: IR Z25 Z20 JUGADOR1 ZAPAS1
28: IR Z20 Z15 JUGADOR1 ZAPAS2
29: DEJAR-OBJETO BIKINI2 Z15 JUGADOR1
30: SACAR-MOCHILA ZAPAS1
31: DEJAR-OBJETO ZAPAS1 Z15 JUGADOR1
32: COGER-OBJETO BIKINI2 Z15 JUGADOR1
33: IR Z15 Z10 JUGADOR1 BIKINI2
34: GIRAR-IZQUIERDA JUGADOR1
35: METER-MOCHILA BIKINI2
36: GIRAR-IZQUIERDA JUGADOR1
37: COGER-OBJETO ROSA2 Z10 JUGADOR1
38: IR Z10 Z15 JUGADOR1 ZAPAS2
39: DEJAR-OBJETO ROSA2 Z15 JUGADOR1
40: COGER-OBJETO ZAPAS1 Z15 JUGADOR1
41: IR Z15 Z20 JUGADOR1 ZAPAS1
42: IR Z20 Z25 JUGADOR1 ZAPAS1
43: GIRAR-DERECHA JUGADOR1
44: IR Z25 Z24 JUGADOR1 ZAPAS1
45: IR Z24 Z23 JUGADOR1 ZAPAS1
46: IR Z23 Z22 JUGADOR1 ZAPAS1
47: IR Z22 Z21 JUGADOR1 ZAPAS2
48: GIRAR-DERECHA JUGADOR1
49: DEJAR-OBJETO ZAPAS1 Z21 JUGADOR1
50: IR Z21 Z16 JUGADOR1 ZAPAS2
51: IR Z16 Z11 JUGADOR1 ZAPAS2
52: IR Z11 Z6 JUGADOR1 ZAPAS2
53: IR Z6 Z1 JUGADOR1 ZAPAS2
54: GIRAR-DERECHA JUGADOR1
55: IR Z1 Z2 JUGADOR1 ZAPAS2
56: IR Z2 Z3 JUGADOR1 BIKINI2
57: GIRAR-IZQUIERDA JUGADOR1
58: GIRAR-IZQUIERDA JUGADOR1
59: COGER-OBJETO ROSA1 Z3 JUGADOR1
```

```
60: IR Z3 Z2 JUGADOR1 ZAPAS2
61: IR Z2 Z1 JUGADOR1 ZAPAS2
62: GIRAR-IZQUIERDA JUGADOR1
63: IR Z1 Z6 JUGADOR1 ZAPAS2
64: IR Z6 Z11 JUGADOR1 ZAPAS2
65: IR Z11 Z16 JUGADOR1 ZAPAS2
66: IR Z16 Z21 JUGADOR1 ZAPAS2
67: GIRAR-IZQUIERDA JUGADOR1
68: DEJAR-OBJETO ROSA1 Z21 JUGADOR1
69: SACAR-MOCHILA BIKINI2
70: DEJAR-OBJETO BIKINI2 Z21 JUGADOR1
71: COGER-OBJETO ZAPAS1 Z21 JUGADOR1
72: METER-MOCHILA ZAPAS1
73: COGER-OBJETO ROSA1 Z21 JUGADOR1
74: IR Z21 Z22 JUGADOR1 ZAPAS1
75: IR Z22 Z23 JUGADOR1 ZAPAS1
76: ENTREGAR-OBJETO ROSA1 Z23 PRINCESA1 JUGADOR1
77: IR Z23 Z24 JUGADOR1 ZAPAS1
78: IR Z24 Z25 JUGADOR1 ZAPAS1
79: GIRAR-IZQUIERDA JUGADOR1
80: IR Z25 Z20 JUGADOR1 ZAPAS1
81: IR Z20 Z15 JUGADOR1 ZAPAS2
82: GIRAR-IZQUIERDA JUGADOR1
83: GIRAR-IZQUIERDA JUGADOR1
84: COGER-OBJETO ROSA2 Z15 JUGADOR1
85: IR Z15 Z20 JUGADOR1 ZAPAS1
86: IR Z20 Z25 JUGADOR1 ZAPAS1
87: GIRAR-DERECHA JUGADOR1
88: IR Z25 Z24 JUGADOR1 ZAPAS1
89: IR Z24 Z23 JUGADOR1 ZAPAS1
90: ENTREGAR-OBJETO ROSA2 Z23 PRINCESA1 JUGADOR1
```

Vemos que aun que no sea demasiado óptimo, se cumple el objetivo.

EJERCICIO 5:

En este problema ahora los personajes tienen un bolsillo con un límite de objetos para almacenar.

Vamos a ver las modificaciones que hemos hecho para que ésto se cumpla:

Hemos añadido dos nuevas funciones, una que lleva el número total de objetos que puede almacenar el bolsillo, y otra que lleva el número de objetos que hay actualmente en el bolsillo.

```
(:functions
  ( distancia ?z1 ?z2 - zona )
  ( distancia_total )

  ( puntos ?obj - objeto ?p - personaje )
  ( puntos_totales )

  ( bolsillo_capacidad_total ?p - personaje )
  ( bolsillo_num_objetos ?p - personaje )

)
```

Ahora, la acción de entregar objeto también es algo diferente.

```
(:action entregar-objeto
  :parameters ( ?obj - objeto ?z - zona ?p - personaje ?j - jugador )
  :precondition ( and (personaje_en ?p ?z) (jugador_en ?j ?z) (mano_ocupada ?obj) (not(es_bikini ?obj))
    (not(es_zapas ?obj)) ( < (bolsillo_num_objetos ?p ) ( bolsillo_capacidad_total ?p ) )
  )
  :effect ( and (mano_vacia) (not(mano_ocupada ?obj)) (personaje_tiene_objeto ?p )
    ( increase ( puntos_totales) ( puntos ?obj ?p ) )
    ( increase ( bolsillo_num_objetos ?p ) 1)
  )
)
```

Como precondition se necesita además que el bolsillo tenga capacidad para almacenar un objeto más, y como efecto incrementa el número de objetos en el bolsillo.

B) Vamos a probar estas modificaciones con nuevos problemas.

En el problema1, la bruja puede almacenar dos objetos en el bolsillo, la princesa almacena uno, y el resto de personajes ninguno. Se buscará un mínimo de 15 puntos.

```
0: GIRAR-IZQUIERDA JUGADOR1
1: GIRAR-IZQUIERDA JUGADOR1
2: IR Z1 Z6 JUGADOR1 ZAPAS2
3: IR Z6 Z11 JUGADOR1 ZAPAS2
4: IR Z11 Z16 JUGADOR1 ZAPAS2
5: IR Z16 Z21 JUGADOR1 ZAPAS2
6: GIRAR-IZQUIERDA JUGADOR1
7: GIRAR-IZQUIERDA JUGADOR1
8: COGER-OBJETO ROSA1 Z21 JUGADOR1
9: IR Z21 Z16 JUGADOR1 ZAPAS2
10: ENTREGAR-OBJETO ROSA1 Z16 BRUJA1 JUGADOR1
11: IR Z16 Z11 JUGADOR1 ZAPAS2
12: IR Z11 Z6 JUGADOR1 ZAPAS2
13: GIRAR-DERECHA JUGADOR1
14: IR Z6 Z7 JUGADOR1 ZAPAS2
15: GIRAR-IZQUIERDA JUGADOR1
16: GIRAR-IZQUIERDA JUGADOR1
17: COGER-OBJETO ZAPAS1 Z7 JUGADOR1
18: IR Z7 Z6 JUGADOR1 ZAPAS2
19: GIRAR-IZQUIERDA JUGADOR1
20: IR Z6 Z11 JUGADOR1 ZAPAS2
21: METER-MOCHILA ZAPAS1
22: IR Z11 Z16 JUGADOR1 ZAPAS2
23: IR Z16 Z21 JUGADOR1 ZAPAS2
24: GIRAR-IZQUIERDA JUGADOR1
25: IR Z21 Z22 JUGADOR1 ZAPAS1
26: IR Z22 Z23 JUGADOR1 ZAPAS1
27: IR Z23 Z24 JUGADOR1 ZAPAS1
28: IR Z24 Z25 JUGADOR1 ZAPAS1
29: GIRAR-IZQUIERDA JUGADOR1
30: IR Z25 Z20 JUGADOR1 ZAPAS1
31: GIRAR-IZQUIERDA JUGADOR1
32: IR Z20 Z19 JUGADOR1 ZAPAS2
33: GIRAR-IZQUIERDA JUGADOR1
34: GIRAR-IZQUIERDA JUGADOR1
35: COGER-OBJETO OSCAR1 Z19 JUGADOR1
36: IR Z19 Z20 JUGADOR1 ZAPAS1
37: GIRAR-DERECHA JUGADOR1
38: IR Z20 Z25 JUGADOR1 ZAPAS1
39: GIRAR-DERECHA JUGADOR1
40: IR Z25 Z24 JUGADOR1 ZAPAS1
41: IR Z24 Z23 JUGADOR1 ZAPAS1
42: GIRAR-IZQUIERDA JUGADOR1
43: DEJAR-OBJETO OSCAR1 Z23 JUGADOR1
44: GIRAR-IZQUIERDA JUGADOR1
45: IR Z23 Z24 JUGADOR1 ZAPAS1
46: GIRAR-IZQUIERDA JUGADOR1
47: COGER-OBJETO BIKINI2 Z24 JUGADOR1
48: GIRAR-IZQUIERDA JUGADOR1
49: IR Z24 Z23 JUGADOR1 ZAPAS1
50: IR Z23 Z22 JUGADOR1 ZAPAS1
```

```
51: IR Z22 Z21 JUGADOR1 ZAPAS2
52: GIRAR-DERECHA JUGADOR1
53: DEJAR-OBJETO BIKINI2 Z21 JUGADOR1
54: SACAR-MOCHILA ZAPAS1
55: DEJAR-OBJETO ZAPAS1 Z21 JUGADOR1
56: COGER-OBJETO BIKINI2 Z21 JUGADOR1
57: IR Z21 Z16 JUGADOR1 ZAPAS2
58: METER-MOCHILA BIKINI2
59: IR Z16 Z11 JUGADOR1 ZAPAS2
60: IR Z11 Z6 JUGADOR1 ZAPAS2
61: IR Z6 Z1 JUGADOR1 ZAPAS2
62: GIRAR-DERECHA JUGADOR1
63: IR Z1 Z2 JUGADOR1 ZAPAS2
64: IR Z2 Z3 JUGADOR1 BIKINI2
65: GIRAR-IZQUIERDA JUGADOR1
66: COGER-OBJETO MANZANA1 Z3 JUGADOR1
67: GIRAR-IZQUIERDA JUGADOR1
68: IR Z3 Z2 JUGADOR1 ZAPAS2
69: IR Z2 Z1 JUGADOR1 ZAPAS2
70: GIRAR-IZQUIERDA JUGADOR1
71: IR Z1 Z6 JUGADOR1 ZAPAS2
72: IR Z6 Z11 JUGADOR1 ZAPAS2
73: IR Z11 Z16 JUGADOR1 ZAPAS2
74: ENTREGAR-OBJETO MANZANA1 Z16 BRUJA1 JUGADOR1
```

Vemos que nuestro plan cumple las condiciones.

Vamos a probar esta vez con el problema2, en el cual todos los personajes tienen 1 hueco, y se buscan 20 puntos.

```
0: GIRAR-IZQUIERDA JUGADOR1
1: GIRAR-IZQUIERDA JUGADOR1
2: IR Z1 Z6 JUGADOR1 ZAPAS2
3: IR Z6 Z11 JUGADOR1 ZAPAS2
4: IR Z11 Z16 JUGADOR1 ZAPAS2
5: IR Z16 Z21 JUGADOR1 ZAPAS2
6: GIRAR-IZQUIERDA JUGADOR1
7: GIRAR-IZQUIERDA JUGADOR1
8: COGER-OBJETO ROSA1 Z21 JUGADOR1
9: IR Z21 Z16 JUGADOR1 ZAPAS2
10: ENTREGAR-OBJETO ROSA1 Z16 BRUJA1 JUGADOR1
11: IR Z16 Z11 JUGADOR1 ZAPAS2
12: IR Z11 Z6 JUGADOR1 ZAPAS2
13: GIRAR-DERECHA JUGADOR1
14: IR Z6 Z7 JUGADOR1 ZAPAS2
15: GIRAR-IZQUIERDA JUGADOR1
16: GIRAR-IZQUIERDA JUGADOR1
17: COGER-OBJETO ZAPAS1 Z7 JUGADOR1
18: IR Z7 Z6 JUGADOR1 ZAPAS2
19: GIRAR-IZQUIERDA JUGADOR1
20: IR Z6 Z11 JUGADOR1 ZAPAS2
21: IR Z11 Z16 JUGADOR1 ZAPAS2
22: IR Z16 Z21 JUGADOR1 ZAPAS2
23: GIRAR-IZQUIERDA JUGADOR1
24: IR Z21 Z22 JUGADOR1 ZAPAS1
25: METER-MOCHILA ZAPAS1
26: IR Z22 Z23 JUGADOR1 ZAPAS1
27: IR Z23 Z24 JUGADOR1 ZAPAS1
28: IR Z24 Z25 JUGADOR1 ZAPAS1
29: GIRAR-IZQUIERDA JUGADOR1
30: IR Z25 Z20 JUGADOR1 ZAPAS1
31: GIRAR-IZQUIERDA JUGADOR1
32: IR Z20 Z19 JUGADOR1 ZAPAS2
33: GIRAR-IZQUIERDA JUGADOR1
34: GIRAR-IZQUIERDA JUGADOR1
35: COGER-OBJETO OSCAR1 Z19 JUGADOR1
36: IR Z19 Z20 JUGADOR1 ZAPAS1
37: GIRAR-DERECHA JUGADOR1
38: IR Z20 Z25 JUGADOR1 ZAPAS1
39: GIRAR-DERECHA JUGADOR1
40: ENTREGAR-OBJETO OSCAR1 Z25 DICAPRIO1 JUGADOR1
41: IR Z25 Z24 JUGADOR1 ZAPAS1
42: COGER-OBJETO BIKINI2 Z24 JUGADOR1
43: IR Z24 Z23 JUGADOR1 ZAPAS1
44: IR Z23 Z22 JUGADOR1 ZAPAS1
45: IR Z22 Z21 JUGADOR1 ZAPAS2
46: GIRAR-DERECHA JUGADOR1
47: DEJAR-OBJETO BIKINI2 Z21 JUGADOR1
48: SACAR-MOCHILA ZAPAS1
49: DEJAR-OBJETO ZAPAS1 Z21 JUGADOR1
50: COGER-OBJETO BIKINI2 Z21 JUGADOR1
51: IR Z21 Z16 JUGADOR1 ZAPAS2
52: METER-MOCHILA BIKINI2
53: IR Z16 Z11 JUGADOR1 ZAPAS2
54: GIRAR-DERECHA JUGADOR1
55: IR Z11 Z12 JUGADOR1 ZAPAS2
56: IR Z12 Z13 JUGADOR1 BIKINI2
57: COGER-OBJETO ALGORITMO1 Z13 JUGADOR1
58: ENTREGAR-OBJETO ALGORITMO1 Z13 PRINCIPE1 JUGADOR1
```

Ha entregado una rosa a la bruja (5 puntos), un oscar a dicaprio (10 puntos) y un algoritmo al príncipe (5 puntos), por lo que vemos que el plan funciona.

EJERCICIO 6.

En este ejercicio, tenemos dos jugadores. Además del total de puntos común, cada jugador deberá obtener una cantidad individual de puntos.

A) Se ha modificado lo siguiente:

En los predicados:

```
(:predicates
  ( connected ?zona1 ?zona2 - zona )
  ( personaje_en ?personaje1 - personaje ?zona1 - zona )
  ( objeto_en ?objeto1 - objeto ?zona1 - zona )
  ( jugador_en ?j - jugador ?zona1 - zona )
  ( mano_vacia ?j - jugador )
  ( mano_ocupada ?obj - objeto ?j - jugador )

  ( personaje_tiene_objeto ?personaje1 - personaje )

  ( jugador_orientado_n ?j - jugador )
  ( jugador_orientado_s ?j - jugador )
  ( jugador_orientado_e ?j - jugador )
  ( jugador_orientado_o ?j - jugador )

  ( zona_orientada_n ?z1 - zona ?z2 - zona )
  ( zona_orientada_s ?z1 - zona ?z2 - zona )
  ( zona_orientada_e ?z1 - zona ?z2 - zona )
  ( zona_orientada_o ?z1 - zona ?z2 - zona )

  ( zona_agua ?z - zona )
  ( zona_precipicio ?z - zona )
  ( zona_bosque ?z )
  ( zona_piedra ?z )
  ( zona_arena ?z )

  ( en_mochila ?obj - objeto ?j - jugador )

  ( es_bikini ?obj - objeto )
  ( es_zapas ?obj - objeto )
```

Como vemos, ahora los predicados relacionados con la mano o la mochila tienen al jugador como argumento. Esto es por que cada jugador tiene su propia mano y su propia mochila.

En las funciones, ahora cada jugador tiene un contador de puntos individuales

```
(:functions
  ( distancia ?z1 ?z2 - zona )
  ( distancia_total )

  ( puntos ?obj - objeto ?p - personaje )
  ( puntos_totales )
  ( puntos_individuales ?j - jugador )

  ( bolsillo_capacidad_total ?p - personaje )
  ( bolsillo_num_objetos ?p - personaje )

)
```

Cada vez que entregamos un objeto, se incrementan ambos contadores, el de puntos comunes y el de puntos individual

```
(:action entregar-objeto
  :parameters ( ?obj - objeto ?z - zona ?p - personaje ?j - jugador )
  :precondition ( and (personaje_en ?p ?z) (jugador_en ?j ?z) (mano_ocupada ?obj ?j) (not(es_bikini ?obj))
    (not(es_zapas ?obj)) ( < (bolsillo_num_objetos ?p ) ( bolsillo_capacidad_total ?p ))
  )
  :effect ( and (mano_vacia ?j ) (not(mano_ocupada ?obj ?j)) (personaje_tiene_objeto ?p )
    ( increase ( puntos_totales) ( puntos ?obj ?p ))
    ( increase ( bolsillo_num_objetos ?p ) 1)
    ( increase (puntos_individuales ?j ) ( puntos ?obj ?p ))
  )
)
```

B) Vamos a poner a prueba estos cambios.

En nuestro problema1, se buscan 30 puntos en total y el jugador1 debe obtener al menos 20.

El plan obtenido es el siguiente:

```
0: GIRAR-IZQUIERDA JUGADOR1
1: GIRAR-IZQUIERDA JUGADOR2
2: GIRAR-IZQUIERDA JUGADOR1
3: IR Z1 Z6 JUGADOR1 ZAPAS2
4: IR Z6 Z11 JUGADOR1 ZAPAS2
5: GIRAR-IZQUIERDA JUGADOR2
6: IR Z1 Z6 JUGADOR2 ZAPAS2
7: IR Z6 Z11 JUGADOR2 ZAPAS2
8: IR Z11 Z16 JUGADOR2 ZAPAS2
9: IR Z11 Z16 JUGADOR1 ZAPAS2
10: IR Z16 Z21 JUGADOR1 ZAPAS2
11: GIRAR-IZQUIERDA JUGADOR1
12: GIRAR-IZQUIERDA JUGADOR1
13: COGER-OBJETO ROSA1 Z21 JUGADOR1
14: IR Z21 Z16 JUGADOR1 ZAPAS2
15: ENTREGAR-OBJETO ROSA1 Z16 BRUJA1 JUGADOR1
16: IR Z16 Z11 JUGADOR1 ZAPAS2
17: IR Z11 Z6 JUGADOR1 ZAPAS2
18: GIRAR-DERECHA JUGADOR1
19: IR Z6 Z7 JUGADOR1 ZAPAS2
20: GIRAR-IZQUIERDA JUGADOR1
21: GIRAR-IZQUIERDA JUGADOR1
22: COGER-OBJETO ZAPAS1 Z7 JUGADOR1
23: IR Z7 Z6 JUGADOR1 ZAPAS2
24: GIRAR-IZQUIERDA JUGADOR1
25: IR Z6 Z11 JUGADOR1 ZAPAS2
26: IR Z11 Z16 JUGADOR1 ZAPAS2
27: IR Z16 Z21 JUGADOR1 ZAPAS2
28: GIRAR-IZQUIERDA JUGADOR1
29: IR Z21 Z22 JUGADOR1 ZAPAS1
30: METER-MOCHILA ZAPAS1 JUGADOR1
31: IR Z22 Z23 JUGADOR1 ZAPAS1
32: IR Z23 Z24 JUGADOR1 ZAPAS1
33: IR Z24 Z25 JUGADOR1 ZAPAS1
34: GIRAR-IZQUIERDA JUGADOR1
35: IR Z25 Z20 JUGADOR1 ZAPAS1
36: GIRAR-IZQUIERDA JUGADOR1
37: IR Z20 Z19 JUGADOR1 ZAPAS2
38: GIRAR-IZQUIERDA JUGADOR1
39: COGER-OBJETO OSCAR1 Z19 JUGADOR1
40: GIRAR-IZQUIERDA JUGADOR1
41: IR Z19 Z20 JUGADOR1 ZAPAS1
42: GIRAR-DERECHA JUGADOR1
43: IR Z20 Z25 JUGADOR1 ZAPAS1
44: ENTREGAR-OBJETO OSCAR1 Z25 DICAPRIO1 JUGADOR1
45: GIRAR-DERECHA JUGADOR1
```

```
46: IR Z25 Z24 JUGADOR1 ZAPAS1
47: GIRAR-IZQUIERDA JUGADOR2
48: GIRAR-IZQUIERDA JUGADOR2
49: IR Z16 Z11 JUGADOR2 ZAPAS2
50: GIRAR-DERECHA JUGADOR2
51: COGER-OBJETO BIKINI2 Z24 JUGADOR1
52: IR Z24 Z23 JUGADOR1 ZAPAS1
53: IR Z23 Z22 JUGADOR1 ZAPAS1
54: IR Z22 Z21 JUGADOR1 ZAPAS2
55: GIRAR-DERECHA JUGADOR1
56: IR Z21 Z16 JUGADOR1 ZAPAS2
57: IR Z16 Z11 JUGADOR1 ZAPAS2
58: DEJAR-OBJETO BIKINI2 Z11 JUGADOR1
59: COGER-OBJETO BIKINI2 Z11 JUGADOR2
60: IR Z11 Z12 JUGADOR2 ZAPAS2
61: METER-MOCHILA BIKINI2 JUGADOR2
62: IR Z12 Z13 JUGADOR2 BIKINI2
63: GIRAR-IZQUIERDA JUGADOR2
64: GIRAR-IZQUIERDA JUGADOR2
65: COGER-OBJETO ALGORITMO1 Z13 JUGADOR2
66: IR Z13 Z12 JUGADOR2 ZAPAS2
67: IR Z12 Z11 JUGADOR2 ZAPAS2
68: DEJAR-OBJETO ALGORITMO1 Z11 JUGADOR2
69: COGER-OBJETO ALGORITMO1 Z11 JUGADOR1
70: IR Z11 Z6 JUGADOR1 ZAPAS2
71: IR Z6 Z1 JUGADOR1 ZAPAS2
72: GIRAR-DERECHA JUGADOR1
73: IR Z1 Z2 JUGADOR1 ZAPAS2
74: ENTREGAR-OBJETO ALGORITMO1 Z2 PROFESOR1 JUGADOR1
75: GIRAR-DERECHA JUGADOR2
76: IR Z11 Z6 JUGADOR2 ZAPAS2
77: IR Z6 Z1 JUGADOR2 ZAPAS2
78: GIRAR-DERECHA JUGADOR2
79: IR Z1 Z2 JUGADOR2 ZAPAS2
80: IR Z2 Z3 JUGADOR2 BIKINI2
81: GIRAR-IZQUIERDA JUGADOR2
82: GIRAR-IZQUIERDA JUGADOR2
83: COGER-OBJETO MANZANA1 Z3 JUGADOR2
84: IR Z3 Z2 JUGADOR2 ZAPAS2
85: ENTREGAR-OBJETO MANZANA1 Z2 PROFESOR1 JUGADOR2
```

Vemos que el jugador1 obtiene 25 puntos, y el jugador2 5, lo que hacen un total de 30, por lo que se cumplen las condiciones.

En nuestro problema2 buscaremos que tanto jugador1 como jugador2 obtengan exactamente 5 puntos.

```
0: GIRAR-IZQUIERDA JUGADOR1
1: GIRAR-IZQUIERDA JUGADOR2
2: GIRAR-IZQUIERDA JUGADOR1
3: GIRAR-IZQUIERDA JUGADOR2
4: IR Z1 Z6 JUGADOR1 ZAPAS2
5: IR Z1 Z6 JUGADOR2 ZAPAS2
6: IR Z6 Z11 JUGADOR1 ZAPAS2
7: IR Z6 Z11 JUGADOR2 ZAPAS2
8: IR Z11 Z16 JUGADOR1 ZAPAS2
9: IR Z11 Z16 JUGADOR2 ZAPAS2
10: IR Z16 Z21 JUGADOR1 ZAPAS2
11: GIRAR-IZQUIERDA JUGADOR1
12: GIRAR-IZQUIERDA JUGADOR1
13: COGER-OBJETO ROSA1 Z21 JUGADOR1
14: IR Z21 Z16 JUGADOR1 ZAPAS2
15: DEJAR-OBJETO ROSA1 Z16 JUGADOR1
16: COGER-OBJETO ROSA1 Z16 JUGADOR2
17: ENTREGAR-OBJETO ROSA1 Z16 BRUJA1 JUGADOR2
18: IR Z16 Z11 JUGADOR1 ZAPAS2
19: IR Z11 Z6 JUGADOR1 ZAPAS2
20: GIRAR-DERECHA JUGADOR1
21: IR Z6 Z7 JUGADOR1 ZAPAS2
22: GIRAR-IZQUIERDA JUGADOR1
23: GIRAR-IZQUIERDA JUGADOR1
24: COGER-OBJETO ZAPAS1 Z7 JUGADOR1
25: IR Z7 Z6 JUGADOR1 ZAPAS2
26: GIRAR-IZQUIERDA JUGADOR1
27: IR Z6 Z11 JUGADOR1 ZAPAS2
28: IR Z11 Z16 JUGADOR1 ZAPAS2
29: IR Z16 Z21 JUGADOR1 ZAPAS2
30: GIRAR-IZQUIERDA JUGADOR1
31: IR Z21 Z22 JUGADOR1 ZAPAS1
32: IR Z22 Z23 JUGADOR1 ZAPAS1
33: IR Z23 Z24 JUGADOR1 ZAPAS1
34: METER-MOCHILA ZAPAS1 JUGADOR1
35: IR Z24 Z25 JUGADOR1 ZAPAS1
36: GIRAR-IZQUIERDA JUGADOR1
37: IR Z25 Z20 JUGADOR1 ZAPAS1
38: GIRAR-IZQUIERDA JUGADOR1
39: IR Z20 Z19 JUGADOR1 ZAPAS2
40: GIRAR-IZQUIERDA JUGADOR1
41: GIRAR-IZQUIERDA JUGADOR1
42: COGER-OBJETO OSCAR1 Z19 JUGADOR1
43: IR Z19 Z20 JUGADOR1 ZAPAS1
44: GIRAR-DERECHA JUGADOR1
45: IR Z20 Z25 JUGADOR1 ZAPAS1
46: GIRAR-DERECHA JUGADOR1
47: IR Z25 Z24 JUGADOR1 ZAPAS1
48: IR Z24 Z23 JUGADOR1 ZAPAS1
49: ENTREGAR-OBJETO OSCAR1 Z23 PRINCESA1 JUGADOR1
```

El jugador2 ha entregado una rosa a la bruja (5 puntos), y el jugador1 ha entregado un oscar a la princesa (5 puntos), por lo que el plan es correcto.

EJERCICIO 7:

En este nuevo escenario los jugadores son de distinto tipo: Uno recoge objetos y se los entrega al otro jugador, que será el que le entregue estos objetos a los personajes.

Los cambios realizados son los siguientes:

Ahora hay dos tipos de jugador: jugador_coge y jugador_entrega.

```
(:types personaje objeto zona jugador orientacion superficie herramientas - object
  princesa principe bruja profesor dicaprio - personaje
  bikini zapatilla manzana rosa algoritmo oro oscar - objeto
  jugador_coge jugador_entrega - jugador
)
```

En coger objeto, sólo el jugador de tipo jugador_coge puede realizar la acción.

```
(:action coger-objeto
  :parameters ( ?obj - objeto ?z - zona ?j - jugador_coge )
  :precondition ( and (mano_vacia ?j) (objeto_en ?obj ?z) (jugador_en ?j ?z) )
  :effect ( and (not(mano_vacia ?j)) (not(objeto_en ?obj ?z)) (mano_ocupada ?obj ?j) )
)
```

Ahora ya no hay una acción que se llame entregar-objeto, hay dos: Entregar-objeto-personaje y entregar-objeto-jugadorentregador.

```
(:action entregar-objeto-personaje
  :parameters ( ?obj - objeto ?z - zona ?p - personaje ?j - jugador_entrega )
  :precondition ( and (personaje_en ?p ?z) (jugador_en ?j ?z) (mano_ocupada ?obj ?j) (not(es_bikini ?obj))
    (not(es_zapas ?obj)) ( < (bolsillo_num_objetos ?p ) ( bolsillo_capacidad_total ?p )) )
  :effect ( and (mano_vacia ?j ) (not(mano_ocupada ?obj ?j)) (personaje_tiene_objeto ?p )
    ( increase ( puntos_totales) ( puntos ?obj ?p ))
    ( increase ( bolsillo_num_objetos ?p ) 1)
  )
)

(:action entregar-objeto-jugadorentregador
  :parameters ( ?obj - objeto ?z - zona ?j1 - jugador_coge ?j2 - jugador_entrega )
  :precondition ( and (jugador_en ?j1 ?z) (jugador_en ?j2 ?z) (mano_ocupada ?obj ?j1) (mano_vacia ?j2) )
  :effect ( and (mano_vacia ?j1 ) (not(mano_ocupada ?obj ?j1)) ( not(mano_vacia ?j2) ) (mano_ocupada ?obj ?j2)
  )
)
```

En la primera acción, sólo el jugador de tipo jugador_entrega puede entregar un objeto a un personaje.

En la segunda acción, el jugador de tipo jugador_coge le entrega un objeto al jugador de tipo jugador_entrega. Como precondiciones, jugador_coge debe tener el objeto en la mano y jugador_entrega debe tener la mano vacía.

El resto de acciones también se han modificado para incluir el argumento de jugador: Girar-derecha, girar-izquierda, ir, etc.

B) Vamos a ver cómo funciona esta nueva modificación.

En el problema1, buscamos que tanto la princesa como el príncipe tengan objeto. El plan es el siguiente.

```
0: GIRAR-IZQUIERDA JUGADOR2
1: GIRAR-IZQUIERDA JUGADOR1
2: GIRAR-IZQUIERDA JUGADOR2
3: IR Z1 Z6 JUGADOR2 ZAPAS2
4: IR Z6 Z11 JUGADOR2 ZAPAS2
5: GIRAR-IZQUIERDA JUGADOR1
6: IR Z1 Z6 JUGADOR1 ZAPAS2
7: GIRAR-IZQUIERDA JUGADOR1
8: IR Z6 Z7 JUGADOR1 ZAPAS2
9: GIRAR-IZQUIERDA JUGADOR1
10: GIRAR-IZQUIERDA JUGADOR1
11: COGER-OBJETO ZAPAS1 Z7 JUGADOR1
12: IR Z7 Z6 JUGADOR1 ZAPAS2
13: GIRAR-IZQUIERDA JUGADOR1
14: IR Z6 Z11 JUGADOR1 ZAPAS2
15: IR Z11 Z16 JUGADOR1 ZAPAS2
16: IR Z16 Z21 JUGADOR1 ZAPAS2
17: GIRAR-IZQUIERDA JUGADOR1
18: IR Z11 Z16 JUGADOR2 ZAPAS2
19: DEJAR-OBJETO ZAPAS1 Z21 JUGADOR1
20: COGER-OBJETO ROSA1 Z21 JUGADOR1
21: IR Z16 Z21 JUGADOR2 ZAPAS2
22: GIRAR-IZQUIERDA JUGADOR2
23: ENTREGAR-OBJETO-JUGADORENTREGADOR ROSA1 Z21 JUGADOR1 JUGADOR2
24: COGER-OBJETO ZAPAS1 Z21 JUGADOR1
25: IR Z21 Z22 JUGADOR1 ZAPAS1
26: IR Z22 Z23 JUGADOR1 ZAPAS1
27: METER-MOCHILA ZAPAS1 JUGADOR1
28: IR Z23 Z24 JUGADOR1 ZAPAS1
29: GIRAR-IZQUIERDA JUGADOR1
30: GIRAR-IZQUIERDA JUGADOR1
31: COGER-OBJETO BIKINI2 Z24 JUGADOR1
32: IR Z24 Z23 JUGADOR1 ZAPAS1
33: IR Z23 Z22 JUGADOR1 ZAPAS1
34: IR Z22 Z21 JUGADOR1 ZAPAS2
35: METER-MOCHILA ROSA1 JUGADOR2
36: ENTREGAR-OBJETO-JUGADORENTREGADOR BIKINI2 Z21 JUGADOR1 JUGADOR2
37: SACAR-MOCHILA ZAPAS1 JUGADOR1
38: DEJAR-OBJETO BIKINI2 Z21 JUGADOR2
39: SACAR-MOCHILA ROSA1 JUGADOR2
40: DEJAR-OBJETO ROSA1 Z21 JUGADOR2
41: ENTREGAR-OBJETO-JUGADORENTREGADOR ZAPAS1 Z21 JUGADOR1 JUGADOR2
42: COGER-OBJETO BIKINI2 Z21 JUGADOR1
43: METER-MOCHILA ZAPAS1 JUGADOR2
44: ENTREGAR-OBJETO-JUGADORENTREGADOR BIKINI2 Z21 JUGADOR1 JUGADOR2
```

```
45: COGER-OBJETO ROSA1 Z21 JUGADOR1
46: DEJAR-OBJETO BIKINI2 Z21 JUGADOR2
47: ENTREGAR-OBJETO-JUGADORENTREGADOR ROSA1 Z21 JUGADOR1 JUGADOR2
48: IR Z21 Z22 JUGADOR2 ZAPAS1
49: IR Z22 Z23 JUGADOR2 ZAPAS1
50: COGER-OBJETO BIKINI2 Z21 JUGADOR1
51: ENTREGAR-OBJETO-PERSONAJE ROSA1 Z23 PRINCESA1 JUGADOR2
52: GIRAR-DERECHA JUGADOR1
53: IR Z21 Z16 JUGADOR1 ZAPAS2
54: IR Z16 Z11 JUGADOR1 ZAPAS2
55: GIRAR-DERECHA JUGADOR1
56: IR Z11 Z12 JUGADOR1 ZAPAS2
57: GIRAR-IZQUIERDA JUGADOR2
58: GIRAR-IZQUIERDA JUGADOR2
59: IR Z23 Z22 JUGADOR2 ZAPAS1
60: IR Z22 Z21 JUGADOR2 ZAPAS2
61: GIRAR-DERECHA JUGADOR2
62: IR Z21 Z16 JUGADOR2 ZAPAS2
63: IR Z16 Z11 JUGADOR2 ZAPAS2
64: GIRAR-DERECHA JUGADOR2
65: IR Z11 Z12 JUGADOR2 ZAPAS2
66: METER-MOCHILA BIKINI2 JUGADOR1
67: IR Z12 Z13 JUGADOR1 BIKINI2
68: GIRAR-IZQUIERDA JUGADOR1
69: GIRAR-IZQUIERDA JUGADOR1
70: COGER-OBJETO ALGORITMO1 Z13 JUGADOR1
71: IR Z13 Z12 JUGADOR1 ZAPAS2
72: ENTREGAR-OBJETO-JUGADORENTREGADOR ALGORITMO1 Z12 JUGADOR1 JUGADOR2
73: SACAR-MOCHILA BIKINI2 JUGADOR1
74: DEJAR-OBJETO ALGORITMO1 Z12 JUGADOR2
75: SACAR-MOCHILA ZAPAS1 JUGADOR2
76: DEJAR-OBJETO ZAPAS1 Z12 JUGADOR2
77: ENTREGAR-OBJETO-JUGADORENTREGADOR BIKINI2 Z12 JUGADOR1 JUGADOR2
78: COGER-OBJETO ALGORITMO1 Z12 JUGADOR1
79: METER-MOCHILA BIKINI2 JUGADOR2
80: ENTREGAR-OBJETO-JUGADORENTREGADOR ALGORITMO1 Z12 JUGADOR1 JUGADOR2
81: IR Z12 Z13 JUGADOR2 BIKINI2
82: ENTREGAR-OBJETO-PERSONAJE ALGORITMO1 Z13 PRINCIPE1 JUGADOR2
```

Como vemos, además de que jugador2 es el único que ha entregado objetos, cada vez que el jugador2 necesita un objeto, el jugador1 lo busca y se lo da.

Vamos a probar otro problema, el problema2, en el cual la bruja tenga un objeto.

```
0: GIRAR-IZQUIERDA JUGADOR2
1: GIRAR-IZQUIERDA JUGADOR1
2: GIRAR-IZQUIERDA JUGADOR2
3: IR Z1 Z6 JUGADOR2 ZAPAS2
4: IR Z6 Z11 JUGADOR2 ZAPAS2
5: GIRAR-IZQUIERDA JUGADOR1
6: IR Z1 Z6 JUGADOR1 ZAPAS2
7: IR Z6 Z11 JUGADOR1 ZAPAS2
8: IR Z11 Z16 JUGADOR1 ZAPAS2
9: IR Z11 Z16 JUGADOR2 ZAPAS2
10: IR Z16 Z21 JUGADOR1 ZAPAS2
11: COGER-OBJETO ROSA1 Z21 JUGADOR1
12: IR Z16 Z21 JUGADOR2 ZAPAS2
13: GIRAR-IZQUIERDA JUGADOR2
14: GIRAR-IZQUIERDA JUGADOR2
15: ENTREGAR-OBJETO-JUGADORENTREGADOR ROSA1 Z21 JUGADOR1 JUGADOR2
16: IR Z21 Z16 JUGADOR2 ZAPAS2
17: ENTREGAR-OBJETO-PERSONAJE ROSA1 Z16 BRUJA1 JUGADOR2
```

Este es mucho más sencillo que el anterior, por lo que los roles de cada jugador se ven más claros: Jugador1 va a por la rosa, se la entrega a jugador2, y jugador2 le entrega la rosa a la bruja.