PRÁCTICA PDDL – TSI

# JULIO FRESNEDA – [JULIOFRESNEDAG@CORREO.UGR.ES](mailto:JULIOFRESNEDAG@CORREO.UGR.ES) – 49215154

## Ejercicio 1

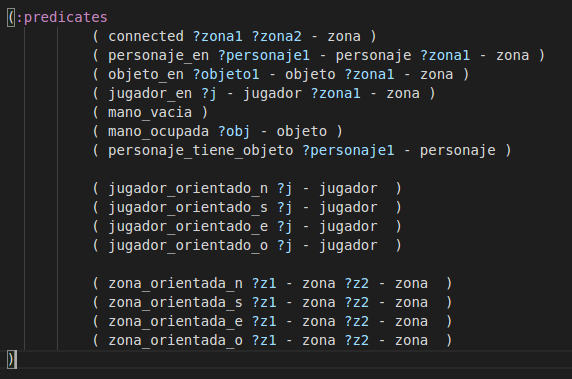
a) Se han representado 4 tipos de objetos: Personaje, objeto, zona y jugador. Los tipos personaje y objeto tienen distintos subtipos:

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

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

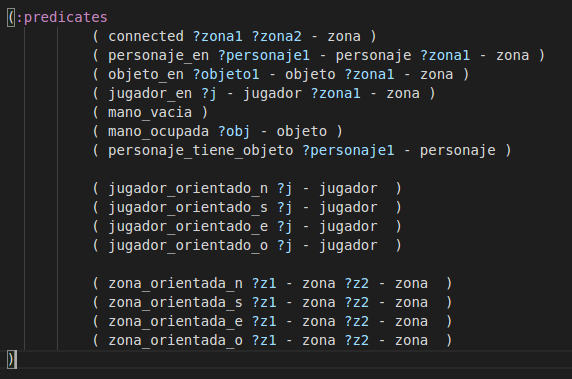
Se han organizado estos subtipos de esta forma por que es más cómodo de programar y más fácil de entender.

b) Los predicados son los siguientes:



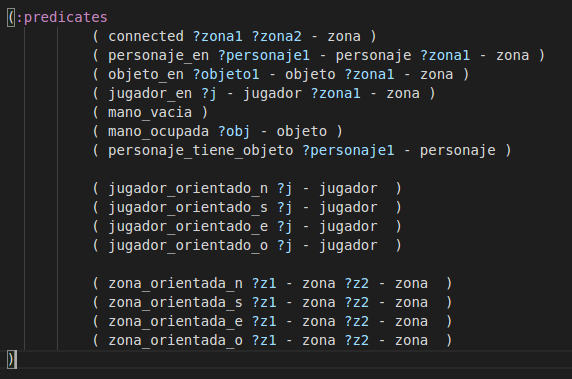
Para conectar unas zonas con otras, se ha usado el predicado ( connected ?zona1 ?zona2 – zona). Este predicado conecta las zonas que están contiguas.

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



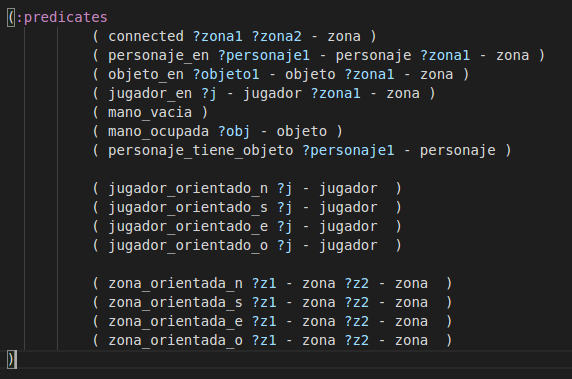
Por ejemplo, si queremos decir que la zona20 está al norte de la zona19, usaríamos el predicado ( zona\_orientada\_n zona20 zona19 ).

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



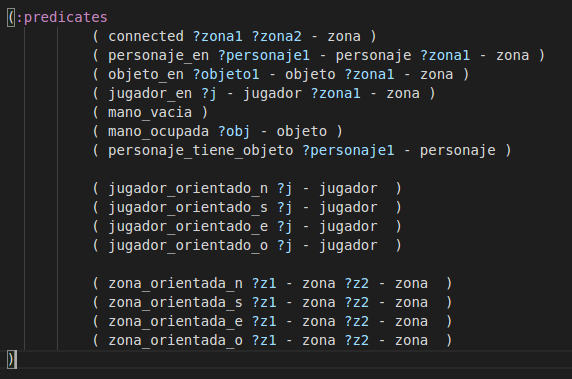
De esta forma cada personaje, objeto y el jugador están asociados a una zona.

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



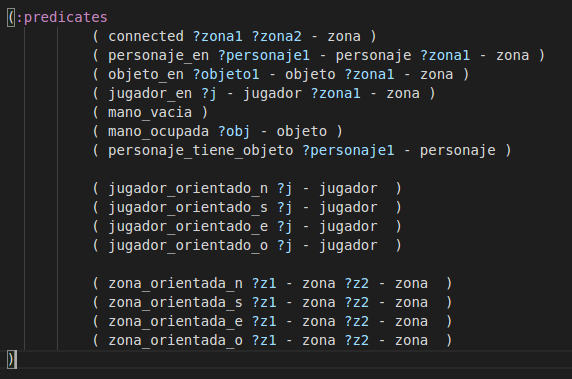
Si un predicado se cumple, el otro no debe cumplirse.

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



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

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

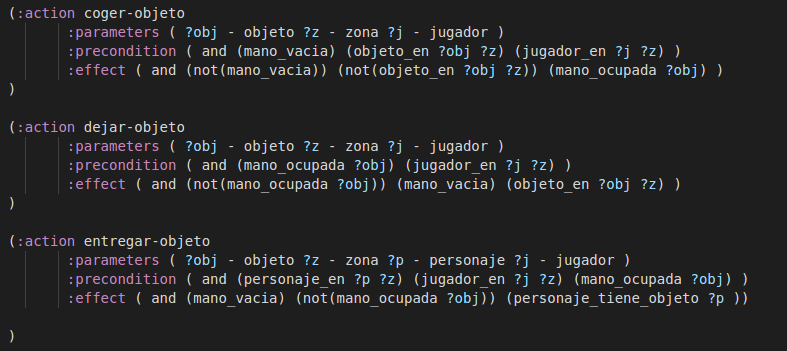


El primer predicado indica que el jugador está orientado hacia el norte; el segundo, hacia el sur; etc.

Cuando uno de estos predicados se cumple, los demás no deben cumplirse.

c) Las acciones son las siguientes:

Coger, dejar y entregar objeto:

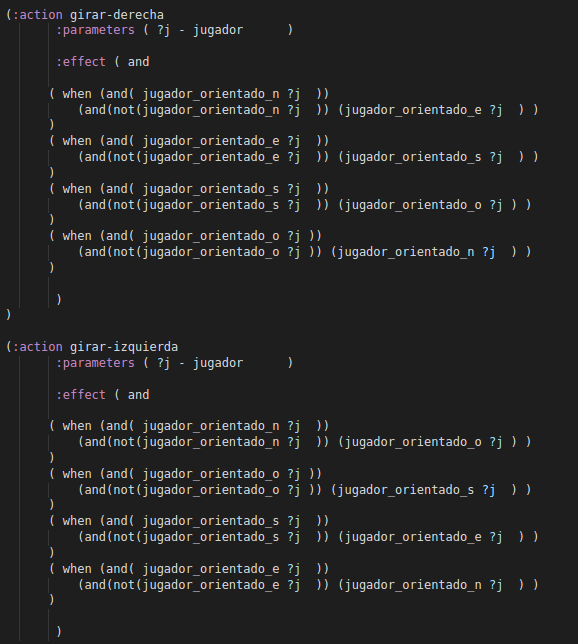


En coger objeto, el jugador coge un objeto del suelo y lo lleva en la mano. Por lo tanto, las precondiciones son que el jugador deba tener la mano vacía y que el objeto esté en la misma zona que el jugador. Los efectos son que el jugador deja de tener la mano vacía y que el objeto ya no está en la zona.

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.

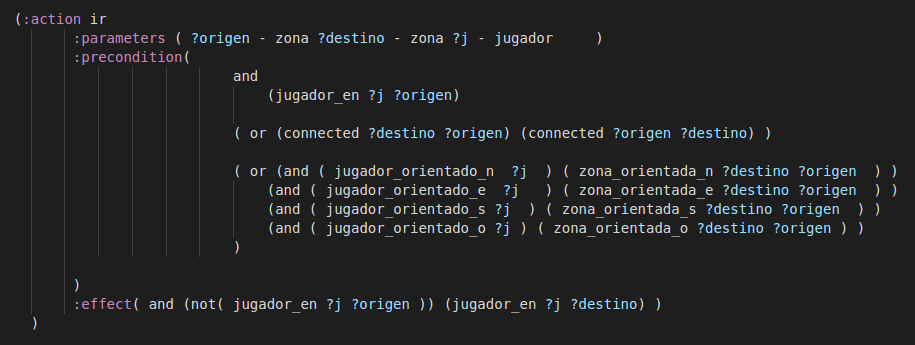
Girar derecha y girar izquierda:



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:



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.

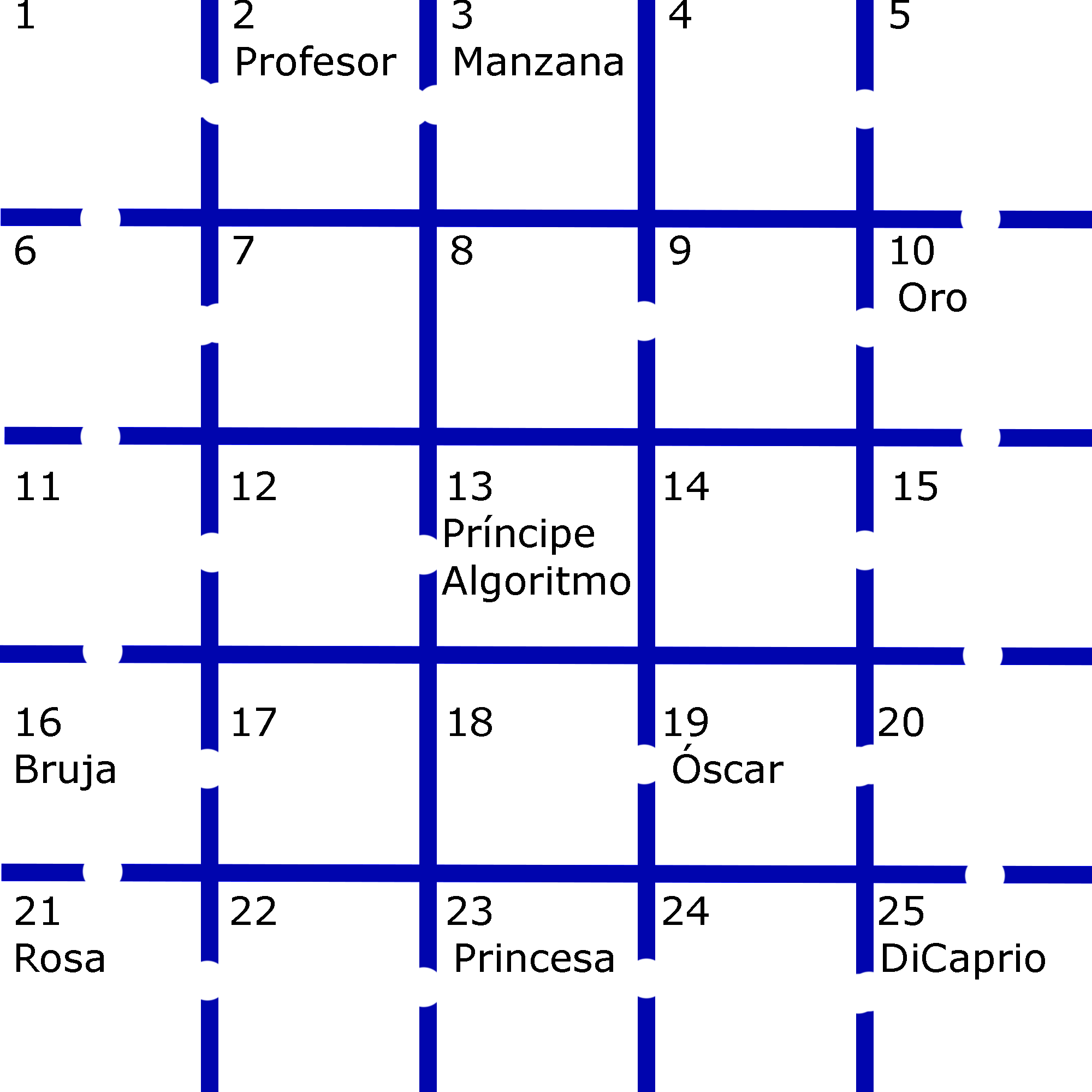
Las precondiciones son que el jugador esté en el origen, que destino y origen estén conectadas, y que el jugador esté orientado en la misma dirección donde está el destino. Por ejemplo, si el destino está al este del origen, el jugador debe estar mirando hacia el este.

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.

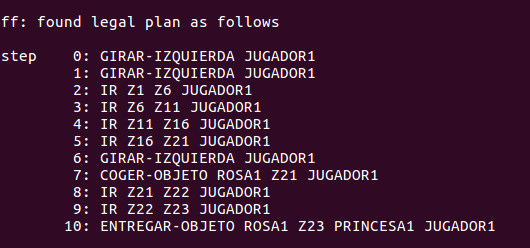
Para ello, nuestro mapa tendrá 25 zonas, con 5 personajes distintos y un mínimo de 5 objetos. El objetivo de este problema será que cada personaje tenga al menos un objeto.

El mapa sería el siguiente:



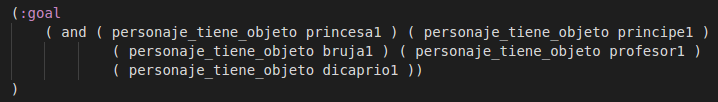
Antes de ver cómo se comporta el plan con el objetivo propuesto, vamos a hacer una pequeña prueba con un objetivo mucho más simple: Que la princesa tenga objeto. Para ello, cargaremos el fichero Ej1problema2.pddl.

El resultado es el siguiente:

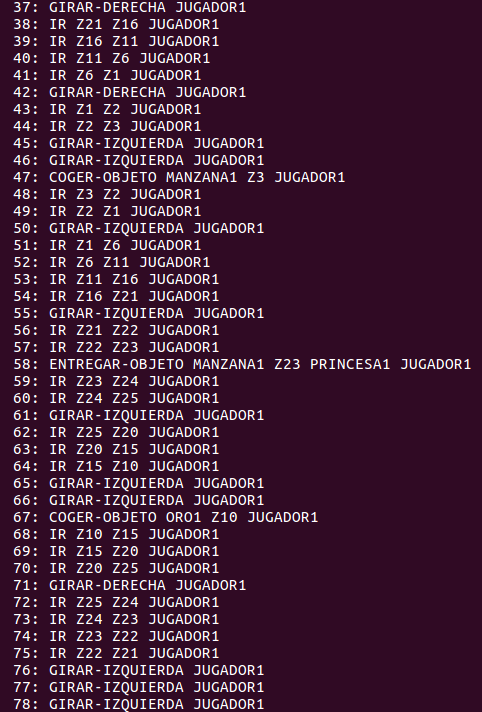
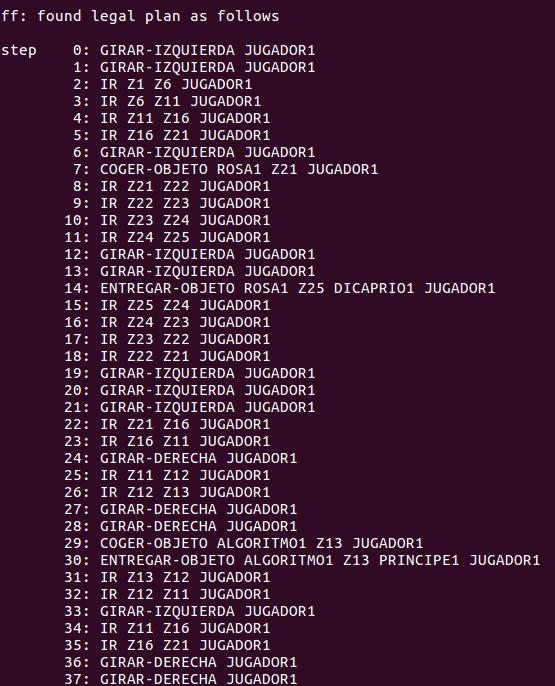


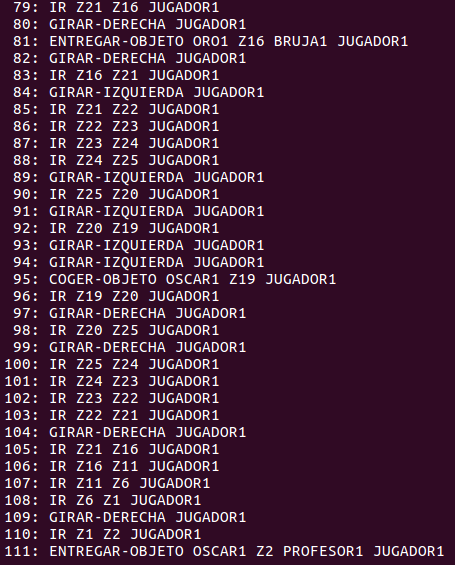
Como vemos, funciona bien.

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



El plan encontrado es el siguiente.





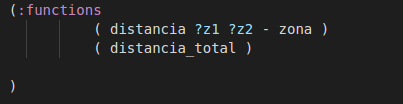
Vemos que es un plan bastante largo, pero si lo seguimos poco a poco podemos comprobar que se cumple, y cada personaje tiene su objeto.

## Ejercicio 2.

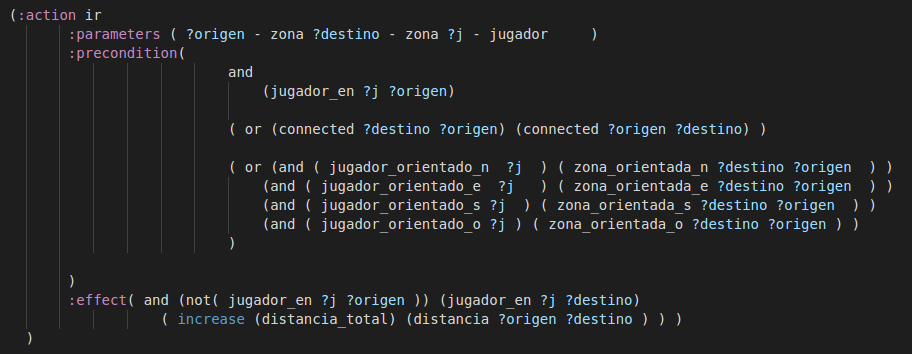
En este nuevo dominio, vamos a asignarle una longitud a cada camino entre zonas contiguas.

a) Las modificaciones del dominio son las siguientes.

Ahora, usaremos dos nuevas funciones:



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.

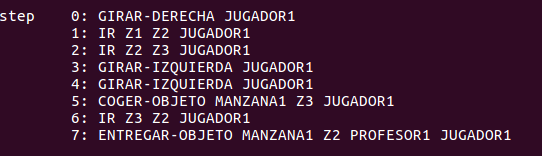
Para comprobar que funciona, vamos a plantear lo siguiente.

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

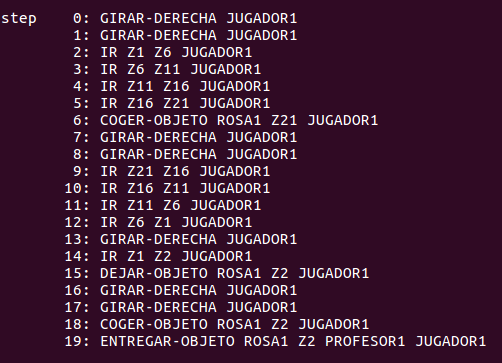


Para ver que funciona, vamos a plantear dos problemas: En ambos se le deberá entregar un objeto al profesor, pero las distancias de cada problema serán diferentes.

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):



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:

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.

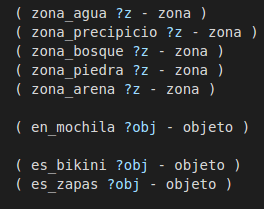
Además, hay dos nuevos objetos: Zapatillas y bikini.

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:

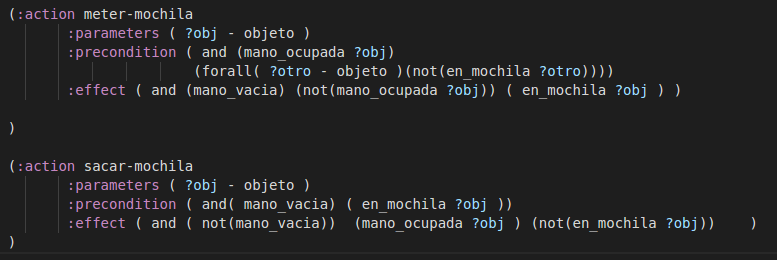


Los cinco primeros predicados clasifican la zona según el tipo que sean.

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.

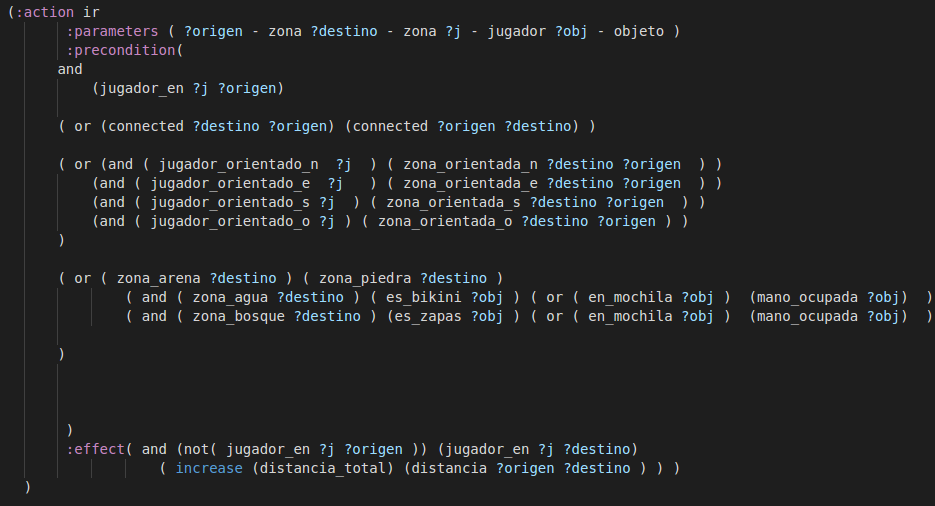
B) Ahora, además de las acciones de coger, dejar y entregar un objeto, están las acciones de meter y sacar un objeto en la mochila. Recordemos que para meter un objeto en la mochila, debemos tener ese objeto en la mano y la mochila debe estar vacía, y para sacar un objeto de la mochila, la mano debe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

Las nuevas acciones son las siguientes:



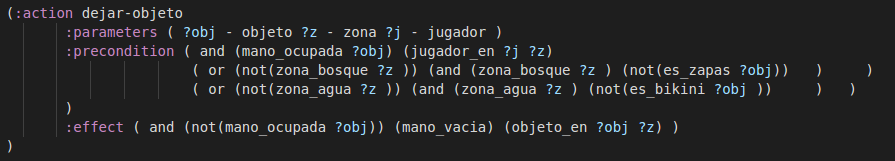
Como vemos, para meter un objeto en la mochila, tenemos como precondición que debemos tener ese objeto en la mano, y que la mochila debe estar vacía. Como efecto vemos que dejamos la mano vacía y la mochila tiene objeto.

Para sacar un objeto de la mochila, es justo al contrario, como precondiciones 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:

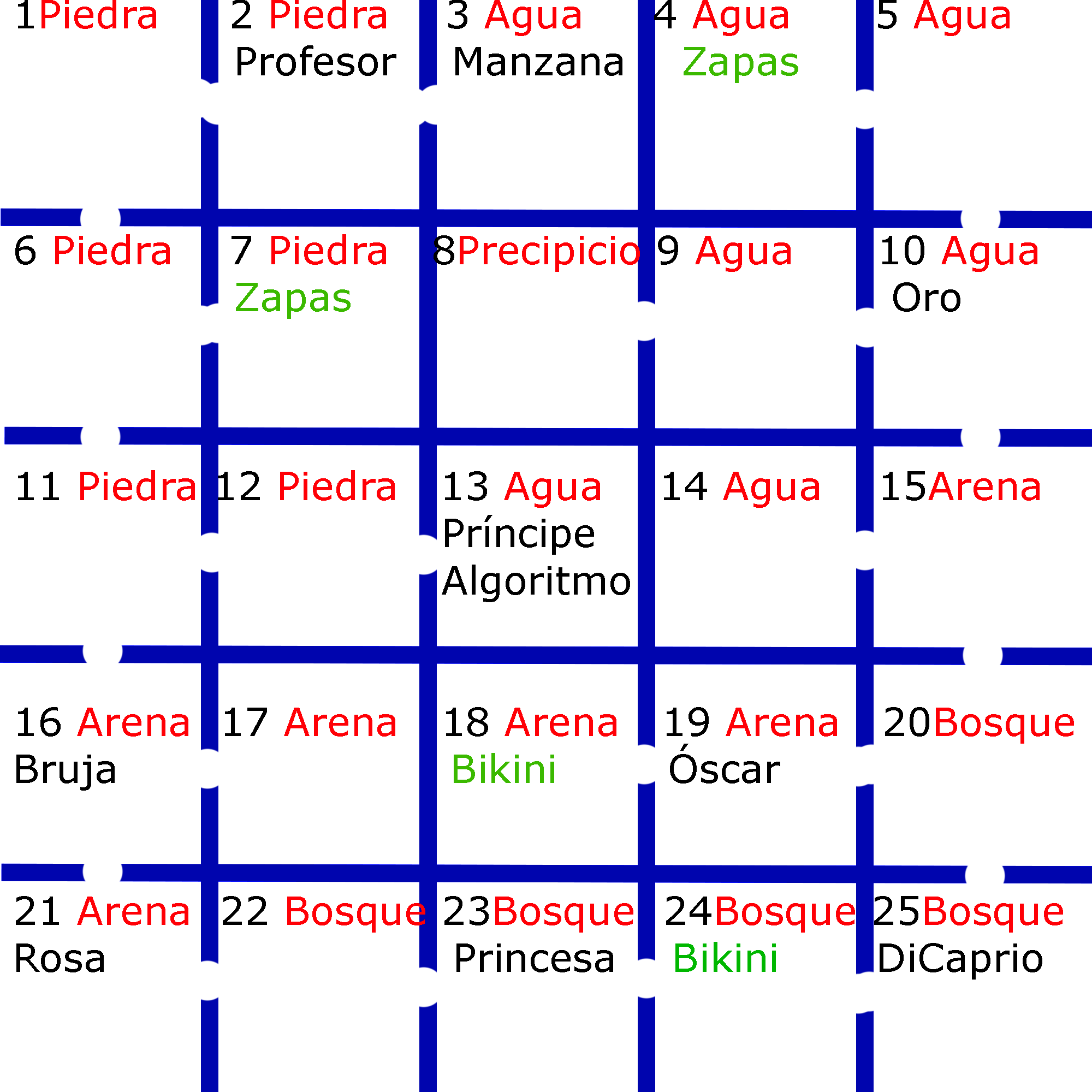
Como vemos en las precondiciones, si el destino es agua, debemos tener un bikini en la mano o en la mochila, y si el destino es bosque, debemos tener unas zapatillas.

En la acción de dejar objeto también hay una diferencia:



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:



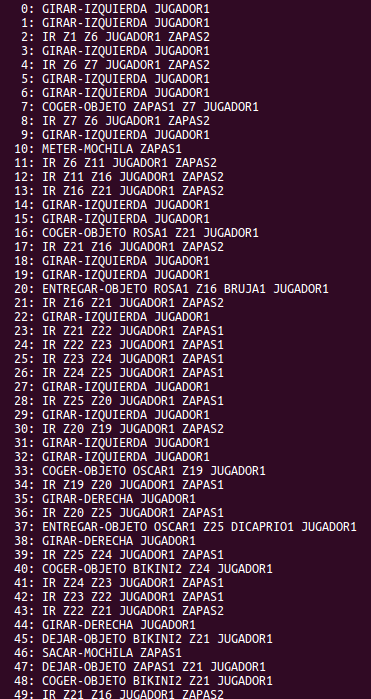
Vamos a ver cómo funciona, con un ejemplo sencillo: Vamos a darle un objeto al príncipe. Pero hay un problema, el príncipe está sobre agua. Vamos a ver cómo se resuelve el problema:

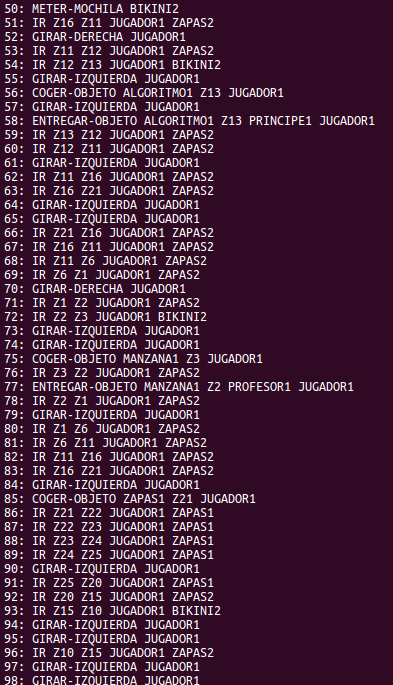
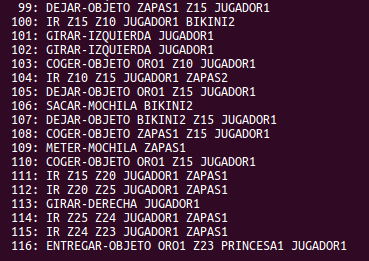
PD: En las acciones de “IR”, siempre a la derecha aparece algún objeto, como zapas2 en este caso. Esto es por que la acción “IR” necesita un objeto como argumento, pero se puede ignorar en la lectura del plan.



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 darselo 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:





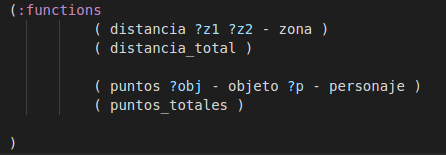
Vemos que es bastante largo, pero si lo seguimos poco a poco vemos que el plan es correcto.

## Ejercicio 4:

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



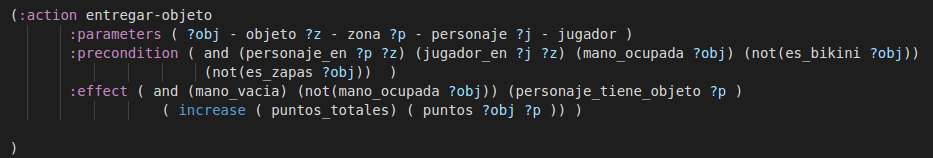
A) Para cumplir ésto se han añadido dos funciones de puntos:



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:



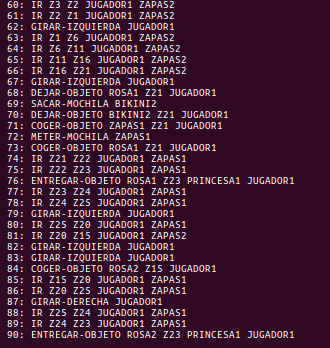
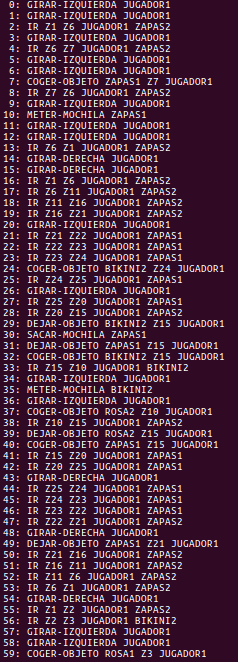
Vemos que en los efectos, le incrementamos a los puntos totales los puntos que hemos conseguido por darle ese objeto a ese personaje.

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



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.

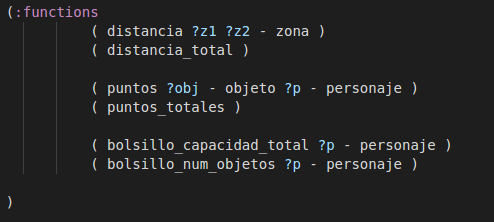


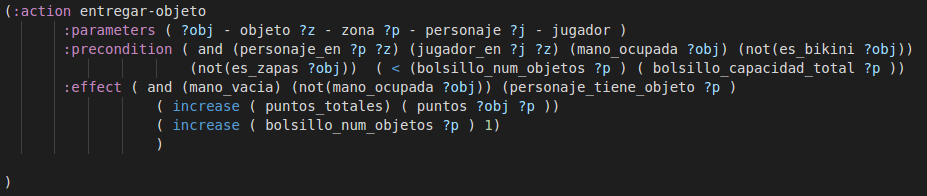
Vemos que aun que no sea demasiado óptimo, se cumple el objetivo.

## 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:

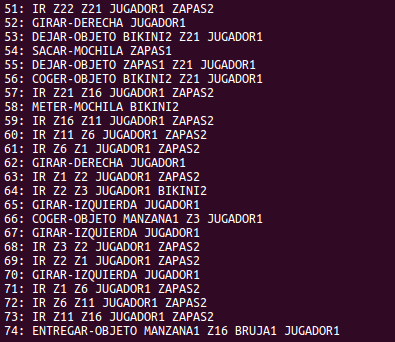
Hemos añadido dos nuevas funciones, una que lleva el número total de objetos que puede almacenar el bolsillo, y otra que lleva el número de objetos que hay actualmente en el bolsillo.

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

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

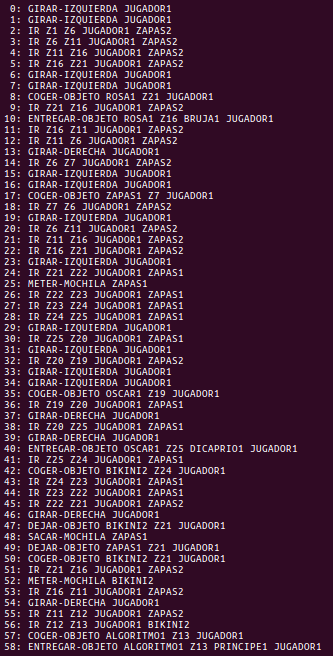
B) Vamos a probar estas modificaciones con nuevos problemas.

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.



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.



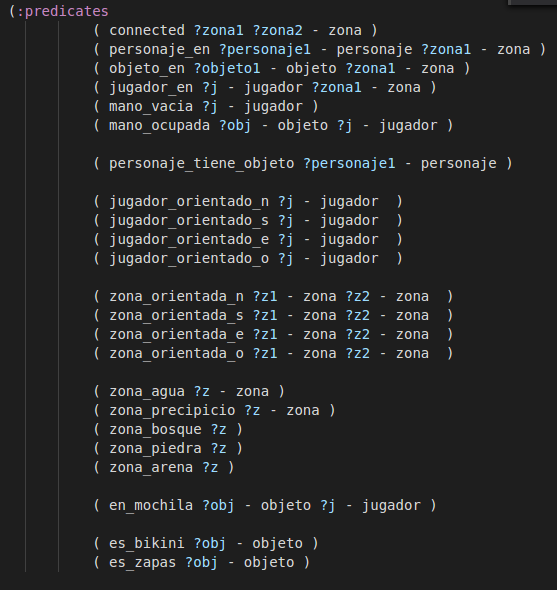
Ha entregado una rosa a la bruja (5 puntos), un oscar a dicaprio (10 puntos) y un algoritmo al príncipe (5 puntos), por lo que vemos que el plan funciona.

## Ejercicio 6.

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

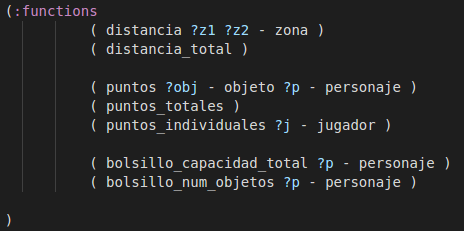
A) Se ha modificado lo siguiente:

En los predicados:

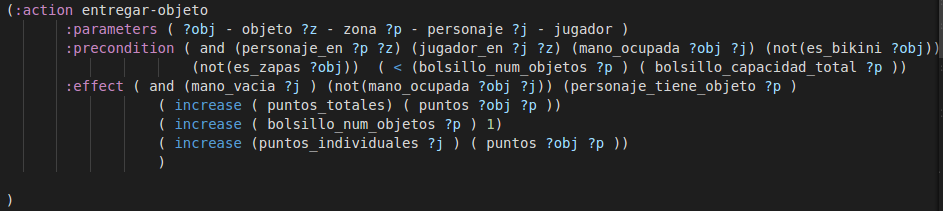


Como vemos, ahora los predicados relacionados con la mano o la mochila tienen al jugador como argumento. Esto es por que cada jugador tiene su propia mano y su propia mochila.

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



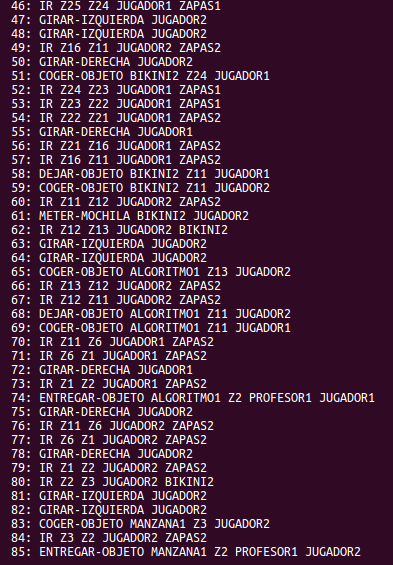
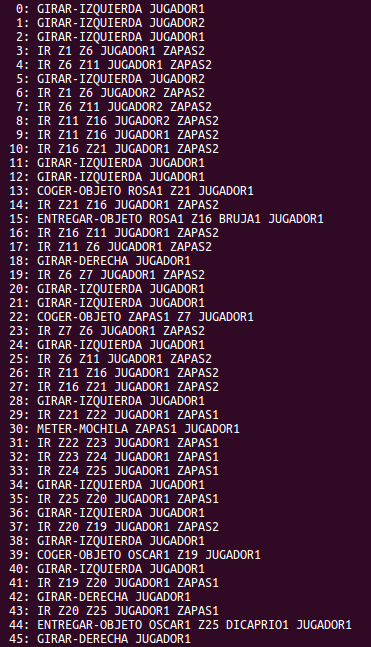
Cada vez que entregamos un objeto, se incrementan ambos contadores, el de puntos comunes y el de puntos individual



B) Vamos a poner a prueba estos cambios.

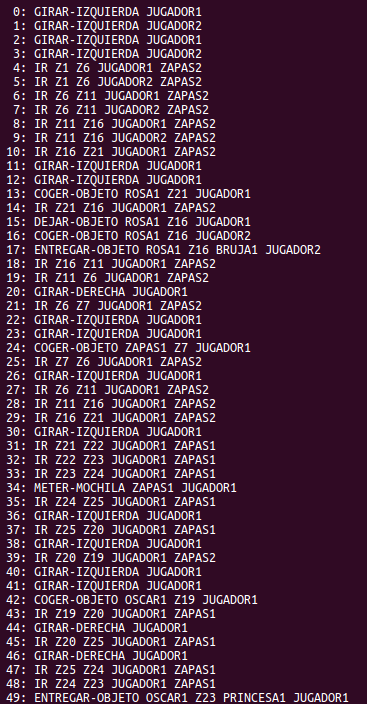
En nuestro problema1, se buscan 30 puntos en total y el jugador1 debe obtener al menos 20.

El plan obtenido es el siguiente:



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.



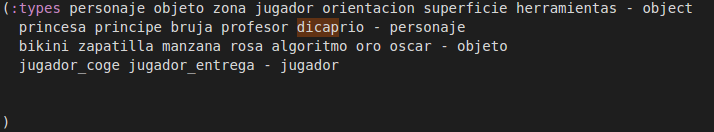
El jugador2 ha entregado una rosa a la bruja (5 puntos), y el jugador1 ha entregado un oscar a la princesa (5 puntos), por lo que el plan es correcto.

## Ejercicio 7:

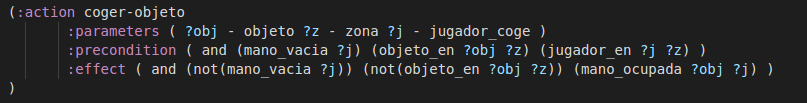
En este nuevo escenario los jugadores son de distinto tipo: Uno recoge objetos y se los entrega al otro jugador, que será el que le entrege estos objetos a los personajes.

Los cambios realizados son los siguientes:

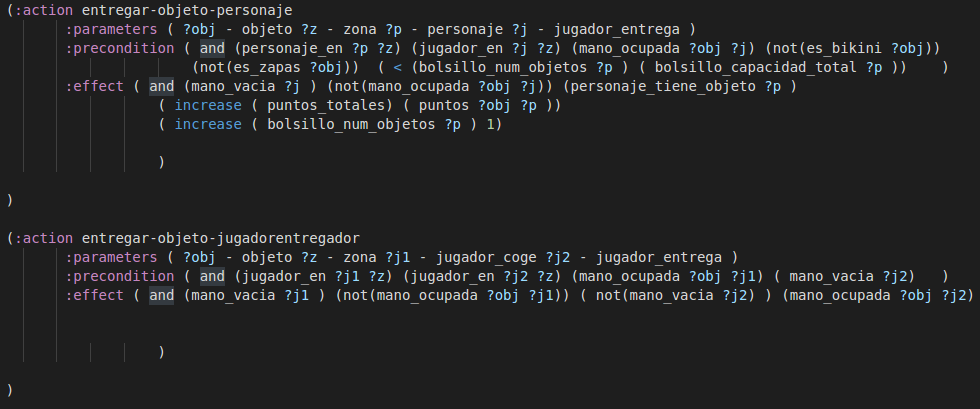
Ahora hay dos tipos de jugador: jugador\_coge y jugador\_entrega.



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



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



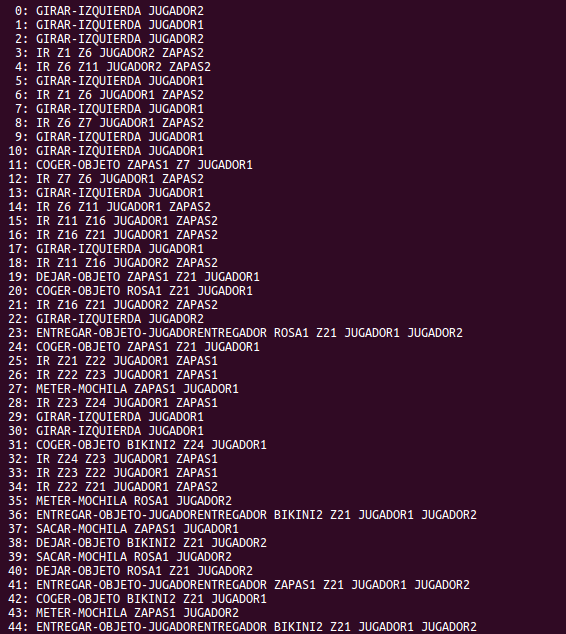
En la primera acción, sólo el jugador de tipo jugador\_entrega puede entregar un objeto a un personaje.

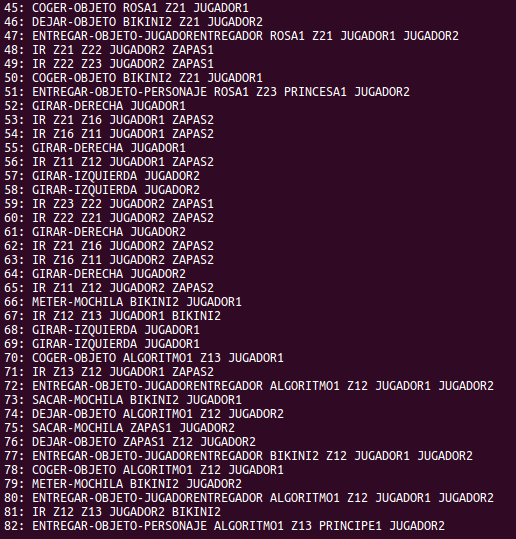
En la segunda acción, el jugador de tipo jugador\_coge le entrega un objeto al jugador de tipo jugador\_entrega. Como precondiciones, jugador\_coge debe tener el objeto en la mano y jugador\_entrega debe tner la mano vacía.

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.

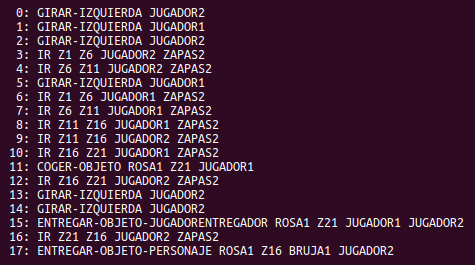
En el problema1, buscamos que tanto la princesa como el príncipe tengan objeto. El plan es el siguiente.





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.



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.