Universidad De Granada

Grado ingeniería informática (2016 – 2017)

Técnicas Sistemas Inteligentes

Práctica 3 | Planificador HTN

Trabajo realizado por: Antonio Miguel Morillo Chica

---

1. Introducción.

En esta practica ahora vamos a usar el planificador htnp creado por el departamento de decsai. Es un planificaco como ff-metric basado en el uso de reglas y con sintaxis pddl. Además tiene un debugguer a diferencia de ff-metric.

2. Ejercicios realizados.

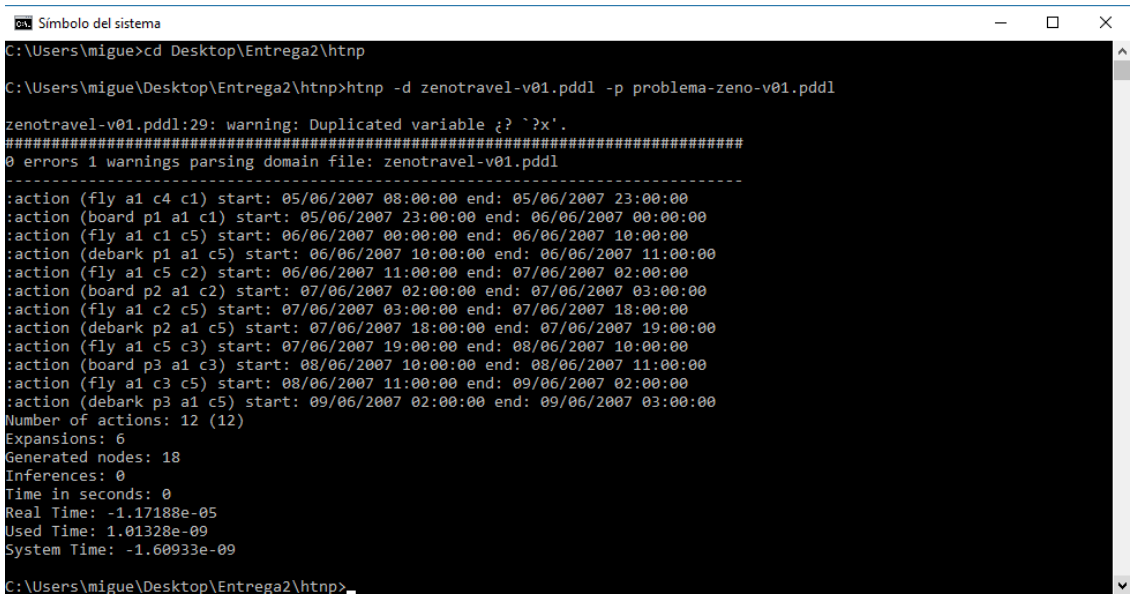
A continuación explicaré como he resuelto los distintos problemas expuestos para la segunda entrega de la practica de planificiación.

- **Ejercicio 1**: *Consiste en transportar 3 personas (inicialmente en las ciudades C1, C2 y C3) a la ciudad C5, considerando que el avión está en la ciudad C4. Se asume al igual que en el problema ejemplo que no hay restricciones de fuel. Comprobar que con el dominio entregado como ejemplo HTNP no encuentra solución y modificar el dominio para que la encuentre.*

Para resolver el problema es necesario tener en cuenta que la persona y el avión podrían no encontrarse en la misma ciudad por lo que habría que desplazar a el avión a la ubicación de la persona.

Esto se resuelve muy fácilmente, lo único que hay que modificar el task de trasportar persona añadiendo un nuevo method en el que se contemplan lo mencionado en el párrafo de arriba.

```
(:method Case3
      :precondition (and
            (at ?p - person ?c1 - city)
            (at ?a - aircraft ?c2 - city)
            (different ?c1 - city ?c2 - city)
      :task (
            (fly ?a ?c2 ?c1)
            (board ?p ?a ?c1 ?c)
            (mover-avion ?a ?c1 ?c)
            (debark ?p ?a ?c)
      )
)
```



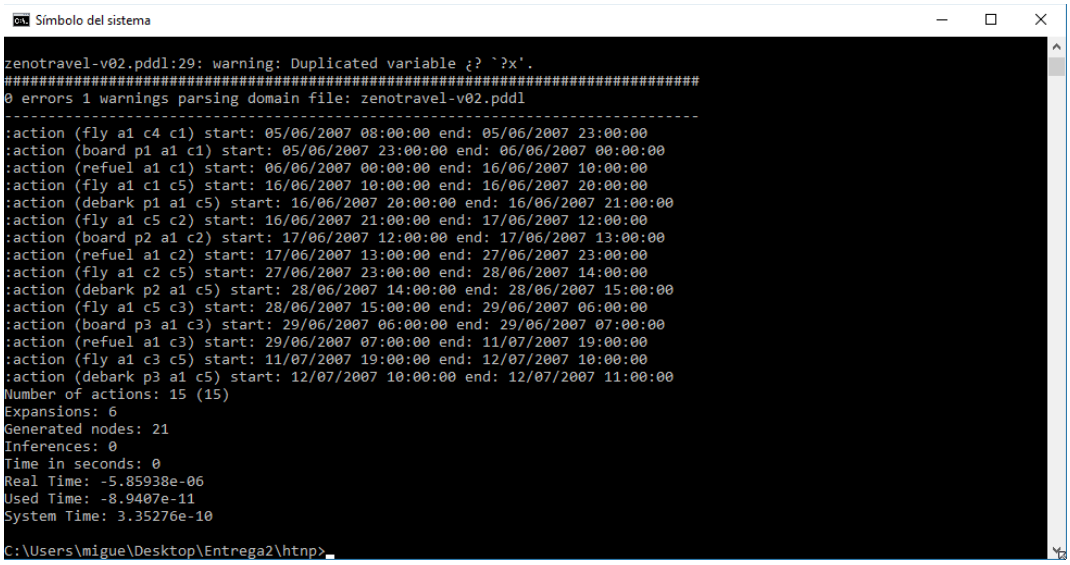
- **Ejercicio 2**: *Consiste en asumir que hay restricciones de fuel. El fuel inicial del avión es de 200 y la capacidad total de 300. Deben contemplarse ahora acciones de repostaje. La situación de partida de personas y avión es la misma que en el problema anterior.*

  Ahora además tendremos que tener en cuenta el fuel necesario para realizar todo el camino, para ello hay que tener en cuenta que el avión debe repostar si ve que se va a quedar sin fuel para poder seguir realizando

  Rescato el domino anterior un método, en mi caso, fuel-insuficiente. Una vez repostado vuela a la ciudad que debe:

```
(:metod fuel-insuficiente
   :precondition (not (hay-fuel ?a ?c1 ?c2)
   :tasks (
      (refuel ?a ?c1)
      (fly ?a ?c1 ?c2)
   )
)
```

  Si no hay fuel suficiente lo que hay que hacer es, claramente, repostar.



- **Ejercicio 3**: *Consiste en considerar acciones de vuelo lento y rápido para tratar de transportar las personas lo más rápido posible con un límite de fuel. En el dominio se tienen que codificar los métodos y tareas de manera que se priorice el uso de acciones de velocidad rápida. El límite de fuel se define con la función (fuel-limit) y es asignado en el estado inicial a 1500. La suma total de fuel gastado en todos los transportes no puede superar 1500 unidades, pero el avión debe viajar siempre lo más rápido posible.*

  Lo primero he hecho ha sido crear el predicado fuel-limit. He modificado el hay fuel de la siguiente forma para que la distancia con combustible para que tenga en cuenta la distancia posible según el fuel:

```
(> (fuel ?a) (*(distance ?c1 ?c2)(slow-burn ?a)))
```

  Tras esto he querido crear métodos para controlar el uso del zoom pero no he podido conseguirlo, aunque he implementado los siguientes:

```
(:derived
    (hay-fuel-rapido ?a - aircraft ?c1 - city ?c2 - city)
    (> (fuel ?a) (* (distance ?c1 ?c2) (slow-burn ?a)))
)

(:method fuel-suficiente-rapido
    :precondition (hay-fuel-rapido ?a ?c1 ?c2)
```

```
      :tasks ( (zoom ?a ?c1 ?c2 ))
  )
```