

# Práctica 3

## Técnicas de los Sistemas Inteligentes

Ignacio Aguilera Martos

1 de junio de 2019

### Índice

1. Ejercicio 1	2
2. Ejercicio 2	2
3. Ejercicio 3	2
4. Ejercicio 4	2

## 1. Ejercicio 1

El primero de los ejercicios nos pide que solucionemos el dominio del que se nos provee para resolver el primero de los problemas HTN. En este caso podemos observar que en la tarea transport-persona hay un caso que no está contemplado, que es precisamente aquel que plantea el problema. Este caso es que el avión y la persona estén en diferentes ciudades para lo que, en primer lugar, el avión se deberá desplazar a la ciudad en la que está la persona y posteriormente ir a la ciudad destino.

Esto en código HTN se refleja poniendo como precondiciones que la persona esté en la ciudad ?c1 y el avión en la ciudad ?c2. Como tareas resultado obtenemos mover el avión de la ciudad ?c2 a ?c1, realizar el embarque, mover el avión de ?c1 a ?c y por último el desembarque.

## 2. Ejercicio 2

Para resolver este segundo ejercicio tenemos que observar que la tarea mover-avion no contempla el caso de que el avión no tenga suficiente combustible para moverse a la siguiente ciudad. Por tanto para resolver el ejercicio debemos añadir un nuevo método que tenga como precondición que no hay suficiente fuel para mover el avión de una ciudad a otra y por tanto tenga como tareas repostar el avión en la ciudad actual y volver a llamar a la tarea mover-avión para que ahora sí se mueva si es que la capacidad de combustible le da la posibilidad de hacerlo.

## 3. Ejercicio 3

En primer lugar he añadido dos predicados (hay-fuel-rapido ?a ?c1 ?c2) y (hay-fuel-lento ?a ?c1 ?c2) que son dos predicados derivados que indican que hay fuel suficiente para ir de forma rápida o lenta por el avión a desde la ciudad 1 a la 2. Además estos predicados derivados tienen como requerimiento, no sólo que haya combustible suficiente para ir de una ciudad a otra, si no además que este incremento de uso de combustible no sobrepase el límite máximo de combustible a usar definido por el problema.

Para darle prioridad a los transportes rápidos he definido en la tarea mover-avion el método fuel-suficiente-lento y fuel-suficiente-rapido. En el rápido se pide como precondición que haya fuel rápido para ir de una ciudad a otra y como tareas tiene la tarea de volar rápido de la ciudad 1 a la 2. En el caso del método fuel lento tiene como precondiciones que haya fuel lento y además que no haya fuel rápido, para que de esta forma se priorice el uso del método de volar rápido antes que el lento siempre que estemos en los requisitos de combustible.

## 4. Ejercicio 4

En primer lugar vamos a ver los cambios que he realizado en el dominio para resolver el problema.

Para poder resolver el primer apartado tenemos que contabilizar el número de pasajeros que tiene un avión dentro. Para ello he definido una nueva función (personas-dentro ?a) que nos

indica el número de personas que tiene dentro el avión ?a. Para ello cada vez que entramos en la primitiva board o debark, como efectos, se incluye incrementar en uno o decrementar en uno esta función dependiendo de si estamos en la acción board o debark. Además para representar la capacidad en personas de un avión he definido otra nueva función (capacidad-personas ?a) que indica la capacidad de personas del avión ?a. Estas dos funciones son empleadas en el método board para comprobar como precondition si hemos excedido o no el límite de personas que pueden embarcar en un avión según su capacidad.

Para el segundo apartado, siguiendo las pistas dadas por el enunciado, he definido un predicado (destino ?p ?c) que nos dice que el destino de la persona ?p es la ciudad ?c. Para poder efectuar el embarque y desembarque de varias personas a la vez tenemos que modificar el comportamiento de la tarea transport-person.

En primer lugar he definido dos nuevas tareas:

- montar-persona: esta tarea tiene como parámetros un avión y una ciudad, de forma que la persona que se va a montar en el avión puede ser inferida. Tiene dos métodos, el primero de ellos es el que se encarga de montar a una persona en un avión. Como preconditiones tiene que la persona esté en la ciudad pasada como parámetro, que el avión esté también en dicha ciudad y que la ciudad en la que se encuentra tanto la persona como el avión no sea el destino de la persona. Las tareas que producen este método son (board ?p ?a ?c1) que se encarga de montar a la persona en el avión y de nuevo la tarea (montar-persona ?a ?c1) que se encarga de volver a comprobar si siguen habiendo personas en la ciudad que puedan montarse en el avión. Por último tenemos el caso base que no tiene preconditiones ni tareas asociadas, de forma que si no se puede aplicar el método anterior no se haga nada y se salga de la recursividad.
- bajar-persona: esta tarea tiene como parámetros una ciudad y un avión de forma que la persona que se va a bajar del avión no viene dada como parámetro y puede ser inferida. Esta tarea, como la anterior tiene dos métodos. El primero de ellos es el que se encarga de bajar a una persona del avión. Como preconditiones tiene que la persona esté dentro del avión, que la ciudad sea el destino de la persona y que el avión esté en la ciudad. De esta forma las tareas que produce el método son (debark ?p ?a ?c1) que indica que la persona se baja del avión y además (bajar-persona ?a ?c1) para que vuelva a comprobar si hay personas que se pueden bajar del avión. Para salir de la recursividad tenemos un método caso base que no tiene preconditiones ni tareas asociadas.

Ahora debemos modificar el funcionamiento de la tarea transport-person. En primer lugar seguimos manteniendo el caso base de que la persona ya esté en la ciudad, en cuyo caso no hacemos nada. Si la persona y el avión están en la misma ciudad llamamos a montar-persona, mover-avion y bajar-persona. De esta forma podemos aplicar las dos tareas nuevas que hemos definido que nos permiten bajar y subir a varias personas. En el caso de que el avión y la persona estén en ciudades distintas primero tenemos que mover el avión a la ciudad de la personas y después aplicar la misma rutina que antes para poder permitir que se suban y bajen varias personas. Por último nos queda un caso que contemplar, este es que la persona ya esté dentro del avión y no sea su ciudad de destino aquella en la que esté el avión. Este caso puede darse pues puede haberse montado previamente en otra ciudad. De esta forma las tareas asociadas serían mover-avion y bajar-persona.

Para el tercer punto vamos a definir dos funciones:

- (maximo-tiempo-vuelo ?a): indica el máximo tiempo de vuelo por avión.
- (tiempo-vuelo ?a): indica el tiempo que ha estado volando un avión.

Con ello ahora cuando llamemos a la primitiva fly o zoom comprobaremos que el avión no ha excedido sus horas máximas de vuelo.

Cabe decir que el límite de fuel ahora pasa de ser global a ser por cada avión.

Vamos a explicar ahora los tres problemas diseñados:

1. El primer problema que he definido impone unas restricciones más fuertes sobre el combustible que sobre las horas máximas de vuelo de cada avión, para poder comprobar que esto funciona. Para ello he posicionado 8 personas y 4 aviones. Además he puesto orígenes comunes para las personas (esto es que hay mas de una persona en alguna ciudad) y destinos comunes (esto es que hay varias personas con el mismo destino). Los aviones están todos posicionados en ciudades diferentes. El primero de ellos tiene una restricción de combustible de 1500, el segundo de 2400, el tercero de 3200 y el cuarto de 4000. De esta forma forzosamente vamos a necesitar dos o tres aviones mínimo para resolver el problema. En este problema la restricción de horas máximas de vuelo para todos los aviones es de 200 horas.
2. El segundo problema toma la misma estructura que el primero, pero en este caso establecemos un uso máximo de combustible para todos los aviones de 5000 de esta forma no representa un gran problema. La restricción temporal en este caso sí se acentúa siendo de 30, 70, 50 y 140 horas para cada uno de los aviones. De esta forma el combustible no será la variable condicionante realmente en el problema, si no el número de horas de vuelo de cada avión.
3. El tercer problema es mucho más grande que los dos anteriores. En este caso tenemos 25 personas y 8 aviones. Hay varias ciudades que comparten personas y todos los aviones están posicionados en ciudades distintas. Las restricciones de combustible son de 7500, 10000, 10000, 1000, 7500, 10000, 10000 y 7500 respectivamente. Las restricciones temporales son 500,400,200,700,500,500,200 y 500. De esta forma proponemos un problema largo y más complejo aunque no podemos retorcer este problema en exceso pues el tiempo de resolución pasa a ser demasiado alto.