



*ugr*

Universidad  
de **Granada**

**Grado en Ingeniería Informática.**

## **Planificación: Práctica 2.**

---

**Nombre de la asignatura:**

Técnicas de los Sistemas Inteligentes.

**Realizado por:**

Néstor Rodríguez Vico



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS  
INFORMÁTICA Y DE TELECOMUNICACIÓN.**

---

Granada, 15 de junio de 2017.

## 1. Ejercicio 1.

Para poder solucionar el problema que se planteaba en el ejercicio 1 sólo he tenido que añadir un método a la tarea *transport-person* que contemple el caso de que el avión no esté en la misma ciudad que la persona. En este caso, debemos transportar el avión a la ciudad de la persona, embarcar a la persona y luego volar a la ciudad destino. El método es el siguiente:

```
(:method Case3 ; El avion y la persona no están en la misma ciudad
  :precondition (and (at ?p - person ?c1 - city)
                     (at ?a - aircraft ?c2 - city)
                     (different ?c1 - city ?c2 - city))
  :tasks ((mover-avion ?a ?c2 ?c1) (board ?p ?a ?c1)
          (mover-avion ?a ?c1 ?c) (debark ?p ?a ?c))
)
```

El resultado de resolver el problema 1 con este dominio es el siguiente:

```
:action (fly a1 c4 c1) start: 05/06/2007 08:00:00 end: 05/06/2007 23:00:00
:action (board p1 a1 c1) start: 05/06/2007 23:00:00 end: 06/06/2007 00:00:00
:action (fly a1 c1 c5) start: 06/06/2007 00:00:00 end: 06/06/2007 10:00:00
:action (debark p1 a1 c5) start: 06/06/2007 10:00:00 end: 06/06/2007 11:00:00
:action (fly a1 c5 c2) start: 06/06/2007 11:00:00 end: 07/06/2007 02:00:00
:action (board p2 a1 c2) start: 07/06/2007 02:00:00 end: 07/06/2007 03:00:00
:action (fly a1 c2 c5) start: 07/06/2007 03:00:00 end: 07/06/2007 18:00:00
:action (debark p2 a1 c5) start: 07/06/2007 18:00:00 end: 07/06/2007 19:00:00
:action (fly a1 c5 c3) start: 07/06/2007 19:00:00 end: 08/06/2007 10:00:00
:action (board p3 a1 c3) start: 08/06/2007 10:00:00 end: 08/06/2007 11:00:00
:action (fly a1 c3 c5) start: 08/06/2007 11:00:00 end: 09/06/2007 02:00:00
:action (debark p3 a1 c5) start: 09/06/2007 02:00:00 end: 09/06/2007 03:00:00
```

## 2. Ejercicio 2.

Para poder solucionar el problema que se planteaba en el ejercicio 2 sólo he tenido que añadir un método a la tarea *mover-avion* que contemple el caso de que el avión no tenga fuel para volar. En este caso lo que debe hacer es llenar el depósito y luego volar. El método es el siguiente:

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

El resultado de resolver el problema 2 con este dominio es el siguiente:

```

:action (fly a1 c4 c1) start: 05/06/2007 08:00:00 end: 05/06/2007 23:00:00
:action (board p1 a1 c1) start: 05/06/2007 23:00:00 end: 06/06/2007 00:00:00
:action (refuel a1 c1) start: 06/06/2007 00:00:00 end: 16/06/2007 10:00:00
:action (fly a1 c1 c5) start: 16/06/2007 10:00:00 end: 16/06/2007 20:00:00
:action (debark p1 a1 c5) start: 16/06/2007 20:00:00 end: 16/06/2007 21:00:00
:action (fly a1 c5 c2) start: 16/06/2007 21:00:00 end: 17/06/2007 12:00:00
:action (board p2 a1 c2) start: 17/06/2007 12:00:00 end: 17/06/2007 13:00:00
:action (refuel a1 c2) start: 17/06/2007 13:00:00 end: 27/06/2007 23:00:00
:action (fly a1 c2 c5) start: 27/06/2007 23:00:00 end: 28/06/2007 14:00:00
:action (debark p2 a1 c5) start: 28/06/2007 14:00:00 end: 28/06/2007 15:00:00
:action (fly a1 c5 c3) start: 28/06/2007 15:00:00 end: 29/06/2007 06:00:00
:action (board p3 a1 c3) start: 29/06/2007 06:00:00 end: 29/06/2007 07:00:00
:action (refuel a1 c3) start: 29/06/2007 07:00:00 end: 11/07/2007 19:00:00
:action (fly a1 c3 c5) start: 11/07/2007 19:00:00 end: 12/07/2007 10:00:00
:action (debark p3 a1 c5) start: 12/07/2007 10:00:00 end: 12/07/2007 11:00:00

```

### 3. Ejercicio 3.

Para este resolver este problema se ha cambiado el predicado (*hay-fuel ?a ?c1 ?c2*) por dos predicados distintos: (*hay-fuel-fast ?a ?c1 ?c2*) y (*hay-fuel-slow ?a ?c1 ?c2*). También se ha añadido la función (*fuel-limit*) para representar la capacidad del avión. Igualmente, el predicado derivado *hay-fuel* se ha eliminado y se han añadido los dos siguientes:

```

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

```

```

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

```

Con estos nuevos cambios la tarea *mover-avion* ha quedado como sigue:

```

(:task mover-avion
  (:method fuel-suficiente-fast
    :precondition (and (hay-fuel-fast ?a ?c1 ?c2)
      (> (fuel-limit) (total-fuel-used)))
    :tasks ((zoom ?a ?c1 ?c2))
  )

  (:method fuel-insuficiente-fast
    :precondition (and (not (hay-fuel-fast ?a ?c1 ?c2))
      (> (fuel-limit) (total-fuel-used)))
  )
)

```

```

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

    (:method fuel-suficiente-slow
      :precondition (and (hay-fuel-slow ?a ?c1 ?c2)
        (> (fuel-limit) (total-fuel-used)))
      :tasks ((fly ?a ?c1 ?c2))
    )

    (:method fuel-insuficiente-slow
      :precondition (and (not (hay-fuel-slow ?a ?c1 ?c2))
        (> (fuel-limit) (total-fuel-used)))
      :tasks ((refuel ?a ?c1) (fly ?a ?c1 ?c2))
    )
  )
)

```

El resultado de resolver el problema 3 con este dominio es el siguiente:

```

:action (refuel a1 c4) start: 05/06/2007 08:00:00 end: 09/06/2007 12:00:00
:action (zoom a1 c4 c1) start: 09/06/2007 12:00:00 end: 09/06/2007 20:00:00
:action (board p1 a1 c1) start: 09/06/2007 20:00:00 end: 09/06/2007 21:00:00
:action (refuel a1 c1) start: 09/06/2007 21:00:00 end: 22/06/2007 09:00:00
:action (zoom a1 c1 c5) start: 22/06/2007 09:00:00 end: 22/06/2007 14:00:00
:action (debark p1 a1 c5) start: 22/06/2007 14:00:00 end: 22/06/2007 15:00:00
:action (refuel a1 c5) start: 22/06/2007 15:00:00 end: 30/06/2007 23:00:00
:action (zoom a1 c5 c2) start: 30/06/2007 23:00:00 end: 01/07/2007 07:00:00
:action (board p2 a1 c2) start: 01/07/2007 07:00:00 end: 01/07/2007 08:00:00
:action (refuel a1 c2) start: 01/07/2007 08:00:00 end: 13/07/2007 20:00:00
:action (zoom a1 c2 c5) start: 13/07/2007 20:00:00 end: 14/07/2007 04:00:00
:action (debark p2 a1 c5) start: 14/07/2007 04:00:00 end: 14/07/2007 05:00:00
:action (refuel a1 c5) start: 14/07/2007 05:00:00 end: 26/07/2007 17:00:00
:action (zoom a1 c5 c3) start: 26/07/2007 17:00:00 end: 27/07/2007 01:00:00
:action (board p3 a1 c3) start: 27/07/2007 01:00:00 end: 27/07/2007 02:00:00
:action (refuel a1 c3) start: 27/07/2007 02:00:00 end: 08/08/2007 14:00:00
:action (zoom a1 c3 c5) start: 08/08/2007 14:00:00 end: 08/08/2007 22:00:00
:action (debark p3 a1 c5) start: 08/08/2007 22:00:00 end: 08/08/2007 23:00:00

```

## 4. Ejercicio 4.

Del ejercicio 4 he conseguido hacer la parte de contabilizar el número de pasajeros. Para ello he añadido dos nuevas funciones: (*maxpasajeros ?a - aircraft*) y (*pasajeros ?a - aircraft*), que representan el número máximo de pasajeros del avión *?a* y el número de pasajeros del avión *?a*, respectivamente. También se han modificado las tareas primitivas, en concreto la tarea de embarque, para contemplar que queda sitio en el avión. Para saber cual es el destino de una persona se ha añadido la función (*destino ?p - person ?c - city*),

la cual representa que las persona  $?p$  va a la ciudad  $?c$ . También se ha realizado las tareas recursivas de *embarque* y *desembarque*. El resultado es el siguiente:

```
(:task board
  :parameters ()
  (:method recurrel
    :precondition (and (at ?p ?c) (at ?a ?c)
      (< (pasajeros ?a) (maxpasajeros ?a))
      (not (destino ?p ?c))))
    :tasks ((board-persona ?p ?a ?c) (board))
  )

  (:method caso-base
    :precondition():tasks()
  )
)

(:durative-action board-persona
  :parameters (?p - person ?a - aircraft ?c - city)
  :duration (= ?duration (boarding-time))
  :condition (and (at ?p ?c) (at ?a ?c)
    (< (pasajeros ?a) (maxpasajeros ?a)))
  :effect (and (not (at ?p ?c)) (in ?p ?a)
    (increase (pasajeros ?a) 1))
)

(:task debark
  :parameters ()
  (:method recurrel
    :precondition (and (in ?p ?a) (at ?a ?c))
    :tasks ((debark-persona ?p ?a ?c) (debark))
  )

  (:method caso-base
    :precondition():tasks()
  )
)
```

```
(:durative-action debark-persona
  :parameters (?p - person ?a - aircraft ?c - city)
  :duration (= ?duration (debarking-time))
  :condition (and (in ?p ?a) (at ?a ?c))
  :effect (and (not (in ?p ?a)) (at ?p ?c)
    (decrease (pasajeros ?a) 1))
)
```

Para probar el funcionamiento del nuevo dominio se han planteado 3 problemas. El primero de ellos tiene 7 personas, p1 está en Granada, p2 está en Bilbao, p3 está en Sevilla, p4 está en Granada, p5 está en Granada, p6 está en Granada y p7 está en Madrid. El avión a1 está en Madrid. El destino de todas las personas es Sevilla. La cantidad máxima de pasajeros del avión *a1* es 20. El resultado obtenido es el siguiente:

```
:action (zoom a1 Madrid Granada) start: 05/06/2007 08:00:00 end: 06/06/2007 05:00:00
:action (board-persona p1 a1 Granada) start: 06/06/2007 05:00:00 end: 06/06/2007 06:00:00
:action (board-persona p4 a1 Granada) start: 06/06/2007 06:00:00 end: 06/06/2007 07:00:00
:action (board-persona p5 a1 Granada) start: 06/06/2007 07:00:00 end: 06/06/2007 08:00:00
:action (board-persona p6 a1 Granada) start: 06/06/2007 08:00:00 end: 06/06/2007 09:00:00
:action (zoom a1 Granada Sevilla) start: 06/06/2007 09:00:00 end: 06/06/2007 22:00:00
:action (debark-persona p1 a1 Sevilla) start: 06/06/2007 22:00:00 end: 06/06/2007 23:00:00
:action (debark-persona p6 a1 Sevilla) start: 06/06/2007 23:00:00 end: 07/06/2007 00:00:00
:action (debark-persona p5 a1 Sevilla) start: 07/06/2007 00:00:00 end: 07/06/2007 01:00:00
:action (debark-persona p4 a1 Sevilla) start: 07/06/2007 01:00:00 end: 07/06/2007 02:00:00
:action (zoom a1 Sevilla Bilbao) start: 07/06/2007 02:00:00 end: 09/06/2007 01:00:00
:action (board-persona p2 a1 Bilbao) start: 09/06/2007 01:00:00 end: 09/06/2007 02:00:00
:action (zoom a1 Bilbao Sevilla) start: 09/06/2007 02:00:00 end: 11/06/2007 01:00:00
:action (debark-persona p2 a1 Sevilla) start: 11/06/2007 01:00:00 end: 11/06/2007 02:00:00
:action (zoom a1 Sevilla Madrid) start: 11/06/2007 02:00:00 end: 12/06/2007 05:00:00
:action (board-persona p7 a1 Madrid) start: 12/06/2007 05:00:00 end: 12/06/2007 06:00:00
:action (zoom a1 Madrid Sevilla) start: 12/06/2007 06:00:00 end: 13/06/2007 09:00:00
:action (debark-persona p7 a1 Sevilla) start: 13/06/2007 09:00:00 end: 13/06/2007 10:00:00
```

El segundo problema de ellos tiene 7 personas, p1, p2, p3, p4, p5, p6 y p7. Todas ellas están en Bilbao. El avión a1 está en Bilbao. El destino de p1 es Bilbao, el de p2 es Sevilla, el de p3 es Granada, el de p4 es Madrid, el de p5 es Madrid, el de p6 es Sevilla y el de p7 es Bilbao. La cantidad máxima de pasajeros del avión *a1* es 30. El resultado obtenido es el siguiente:



```

:action (board-persona p2 a1 Bilbao) start: 05/06/2007 08:00:00 end: 05/06/2007 09:00:00
:action (board-persona p3 a1 Bilbao) start: 05/06/2007 09:00:00 end: 05/06/2007 10:00:00
:action (board-persona p4 a1 Bilbao) start: 05/06/2007 10:00:00 end: 05/06/2007 11:00:00
:action (board-persona p5 a1 Bilbao) start: 05/06/2007 11:00:00 end: 05/06/2007 12:00:00
:action (board-persona p6 a1 Bilbao) start: 05/06/2007 12:00:00 end: 05/06/2007 13:00:00
:action (zoom a1 Bilbao Sevilla) start: 05/06/2007 13:00:00 end: 07/06/2007 12:00:00
:action (debark-persona p2 a1 Sevilla) start: 07/06/2007 12:00:00 end: 07/06/2007 13:00:00
:action (debark-persona p6 a1 Sevilla) start: 07/06/2007 13:00:00 end: 07/06/2007 14:00:00
:action (debark-persona p5 a1 Sevilla) start: 07/06/2007 14:00:00 end: 07/06/2007 15:00:00
:action (debark-persona p4 a1 Sevilla) start: 07/06/2007 15:00:00 end: 07/06/2007 16:00:00
:action (debark-persona p3 a1 Sevilla) start: 07/06/2007 16:00:00 end: 07/06/2007 17:00:00
:action (board-persona p3 a1 Sevilla) start: 07/06/2007 17:00:00 end: 07/06/2007 18:00:00
:action (board-persona p4 a1 Sevilla) start: 07/06/2007 18:00:00 end: 07/06/2007 19:00:00
:action (board-persona p5 a1 Sevilla) start: 07/06/2007 19:00:00 end: 07/06/2007 20:00:00
:action (zoom a1 Sevilla Granada) start: 07/06/2007 20:00:00 end: 08/06/2007 09:00:00
:action (debark-persona p5 a1 Granada) start: 08/06/2007 09:00:00 end: 08/06/2007 10:00:00
:action (debark-persona p4 a1 Granada) start: 08/06/2007 10:00:00 end: 08/06/2007 11:00:00
:action (debark-persona p3 a1 Granada) start: 08/06/2007 11:00:00 end: 08/06/2007 12:00:00
:action (board-persona p4 a1 Granada) start: 08/06/2007 12:00:00 end: 08/06/2007 13:00:00
:action (zoom-persona p5 a1 Granada) start: 08/06/2007 13:00:00 end: 08/06/2007 14:00:00
:action (zoom a1 Granada Madrid) start: 08/06/2007 14:00:00 end: 09/06/2007 11:00:00
:action (debark-persona p5 a1 Madrid) start: 09/06/2007 11:00:00 end: 09/06/2007 12:00:00
:action (debark-persona p4 a1 Madrid) start: 09/06/2007 12:00:00 end: 09/06/2007 13:00:00

```

El segundo problema de ellos tiene 5 personas, p1 está en Barcelona, p2 está en Madrid, p3 está en Granada, p4 está en Sevilla y p5 está en Bilbao. El avión a1 está en Barcelona. El destino de todas las personas es Barcelona. La cantidad máxima de pasajeros del avión *a1* es 20. El resultado obtenido es el siguiente:

```

:action (zoom a1 Barcelona Madrid) start: 05/06/2007 08:00:00 end: 06/06/2007 15:00:00
:action (board-persona p2 a1 Madrid) start: 06/06/2007 15:00:00 end: 06/06/2007 16:00:00
:action (zoom a1 Madrid Barcelona) start: 06/06/2007 16:00:00 end: 07/06/2007 23:00:00
:action (debark-persona p2 a1 Barcelona) start: 07/06/2007 23:00:00 end: 08/06/2007 00:00:00
:action (zoom a1 Barcelona Granada) start: 08/06/2007 00:00:00 end: 09/06/2007 19:00:00
:action (board-persona p3 a1 Granada) start: 09/06/2007 19:00:00 end: 09/06/2007 20:00:00
:action (zoom a1 Granada Barcelona) start: 09/06/2007 20:00:00 end: 11/06/2007 15:00:00
:action (debark-persona p3 a1 Barcelona) start: 11/06/2007 15:00:00 end: 11/06/2007 16:00:00
:action (zoom a1 Barcelona Sevilla) start: 11/06/2007 16:00:00 end: 13/06/2007 20:00:00
:action (board-persona p4 a1 Sevilla) start: 13/06/2007 20:00:00 end: 13/06/2007 21:00:00
:action (zoom a1 Sevilla Barcelona) start: 13/06/2007 21:00:00 end: 16/06/2007 01:00:00
:action (debark-persona p4 a1 Barcelona) start: 16/06/2007 01:00:00 end: 16/06/2007 02:00:00
:action (zoom a1 Barcelona Bilbao) start: 16/06/2007 02:00:00 end: 17/06/2007 09:00:00
:action (board-persona p5 a1 Bilbao) start: 17/06/2007 09:00:00 end: 17/06/2007 10:00:00
:action (zoom a1 Bilbao Barcelona) start: 17/06/2007 10:00:00 end: 18/06/2007 17:00:00
:action (debark-persona p5 a1 Barcelona) start: 18/06/2007 17:00:00 end: 18/06/2007 18:00:00

```

He intentado hacer la parte de las distancias, pero no lo he conseguido. Lo que he hecho ha sido crear un predicado derivado de la siguiente forma:

```

(: derived
  (distancia ?c1 - city ?c2 - city ?d - number)
  (or (bind ?d (distance ?c1 ?c2)) (bind ?d (distance ?c2 ?c1)))
)

```

Este predicado se usa para poder obtener la distancia de forma recíproca. También se han usado funciones que llaman a código Python para realizar operaciones aritméticas. Lo que he conseguido hacer está en el fichero zip llamado *Ej4-fallido.zip*.