



Dominio “Transporte”

Ejercicio 1: Dominio proposicional

En este dominio tenemos:

- **Tres ciudades:** CIUDAD 1, CIUDAD 2 Y CIUDAD 3
- **localizaciones:** casas, estaciones de tren y aeropuertos,
- **medios de transporte:** furgonetas, trenes y aviones,
- **conductores y paquetes** que tienen que ser transportados de una ciudad a otra.

Inicialmente los paquetes están ubicados en una localización de una ciudad y el objetivo es transportar todos los paquetes a las localizaciones de sus ciudades destino.

Los paquetes se pueden cargar en furgonetas, trenes o aviones para ser transportados de una ciudad a otra. No existe límite en la capacidad de los medios de transporte. No se necesita una persona para cargar y descargar paquetes.

Las **furgonetas** se pueden mover entre **cualquier par de puntos de una misma ciudad**. Los **trenes** pueden moverse entre **estaciones de tren de la misma ciudad o distintas ciudades**. Los **aviones** sólo se mueven **entre aeropuertos de distintas ciudades**.

Una **furgoneta necesita un conductor** para poder moverse; además, **los conductores pueden desplazarse CAMINANDO de una localización a otra de la misma ciudad** (nunca entre ciudades distintas). Siempre existe conexión entre cualquier par de localizaciones de una ciudad. En la figura, las conexiones entre localizaciones aparecen señaladas con líneas discontinuas. Dichas conexiones indican los trayectos que pueden realizar los conductores, bien conduciendo las furgonetas, o bien caminando.

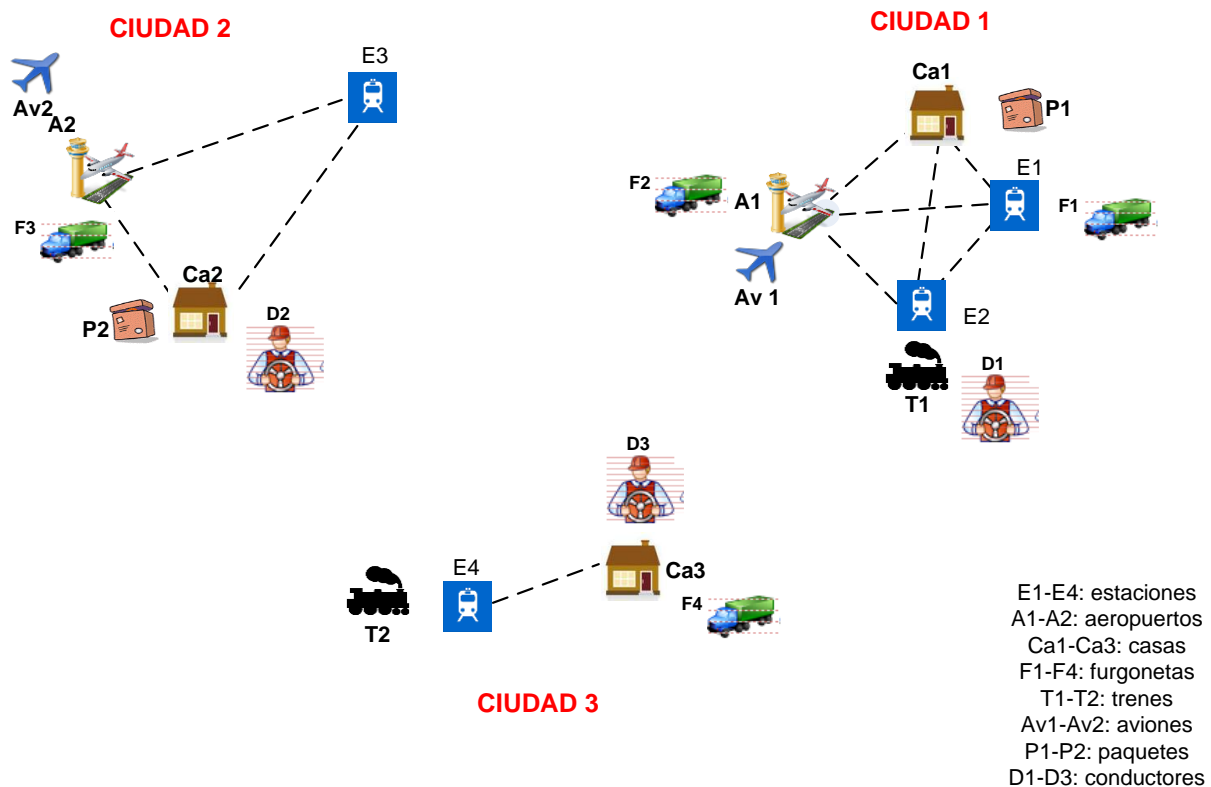
Ejemplo de **situación inicial**: la que se indica en la figura.

El objetivo final es **llevar los paquetes P1 y P2 a la casa Ca3**.

Se pide:

- Definir el dominio correspondiente con las posibles acciones u operadores a aplicar utilizando el lenguaje PDDL.
- Definir la instancia del problema, describiendo los predicados que definen la situación inicial y el objetivo del problema.
- Ejecutar los planificadores FF, LPG, LPG con la opción *-timesteps* y OPTIC, y comprobar si el plan ejecutado resuelve el problema especificado.

- d) Especificar otras instancias de problema, incluyendo nuevos paquetes, furgonetas, trenes, aviones y/o conductores. No modificar la estructura de las ciudades (mantener el número y tipo de localizaciones en cada ciudad). Para crear nuevas instancias hay que cambiar el estado inicial y/o final.





Ejercicio 2: Dominio temporal

- Conductores:
 - subir y bajar de un medio de transporte, 2 unidades de tiempo
 - Paquetes:
 - Definir peso de los paquetes
 - Establecer un tiempo de carga/descarga proporcional al peso del paquete
 - Aviones-Aeropuertos:
 - Definir tiempo de recorrido en avión entre aeropuertos
 - Trenes-Estaciones:
 - Definir tiempo de recorrido en tren entre estaciones
 - Caminos entre localizaciones de una ciudad:
 - Definir distancia de los caminos
 - Establecer tiempo de recorrido (caminando) proporcional a la distancia y en función de la velocidad a la que anda el conductor
 - Establecer tiempo de recorrido (en furgoneta) proporcional a la distancia y la velocidad de la furgoneta
-
- 1) Definir el dominio temporal. Indicar si habéis realizado algún cambio respecto a la codificación STRIPS.
 - 2) Definir la instancia del problema del primer apartado.
 - 3) Ejecutar los planificadores LPG y OPTIC, y comprobar si los planes devueltos son correctos y resuelven el problema especificado. Comparar las soluciones obtenidas.
 - 4) Generar y ejecutar otras instancias de problema cambiando la situación inicial y final.



Ejercicio 3: Dominio con recursos numéricos

Una vez visto el manejo de los *fluents* en PDDL2.1, y sus posibles modificaciones (*assign*, *increase*, *decrease*), realiza las siguientes tareas:

- 1) Define una variable numérica que represente el consumo de un recurso, tal que el consumo del recurso sea inversamente proporcional al consumo de tiempo; es decir, a menos tiempo, más consumo del recurso.
- 2) Define dos versiones del dominio:
 - a. Una versión donde el recurso no sea renovable, es decir, no se puede recargar
 - b. Una versión donde el recurso es renovable
- 3) Ejecuta dos instancias de cada nuevo dominio con LPG. Recuerda que la función a optimizar ahora será: `(:metric minimize (<recurso-definido>))`
- 4) Ejecuta las mismas instancias de cada dominio pero minimizando el tiempo: `(:metric minimize (total-time))`
- 5) Compara los resultados obtenidos.



Ejercicio 4. Desarrollo parcial de un árbol POP (sin tiempo ni recursos)

Dado el problema del ejercicio 1, escoge un solo objetivo (llevar un paquete de un punto a otro del mapa), y aplica tres iteraciones del algoritmo de un planificador POP. Cosas a tener en cuenta:

- 1) En la primera iteración, escoge un nodo del nivel 1 del árbol. Puedes aplicar una de las heurísticas POP o bien escoger un nodo aleatoriamente. En la segunda iteración, escoge un nodo del nivel 2 del árbol. Igualmente, puedes aplicar una heurística POP o bien escoger aleatoriamente un nodo del árbol. Idem para el nivel 3 del árbol POP.
- 2) En cada nodo sucesor, indica el *flaw* que se escoge para su resolución (open goal, amenaza o instanciación de variable). También se puede optar por trabajar con un dominio totalmente instanciado (*grounded*) en cuyo caso solo se escogerá entre una precondición pendiente de resolver o una amenaza, si existiese.
- 3) Indica los enlaces causales y relaciones de orden de cada nodo.

Comenta dónde crees que se encontrarían las dificultades en el desarrollo y resolución de dicho problema.



Ejercicio 5: Graphplan (sin tiempo ni recursos)

Partiendo del problema del primer apartado, y dados los dos objetivos definidos en dicha instancia:

- a) Tener paquete 1 (P1) en la casa 3 (Ca3)
- b) Tener paquete 2 (P2) en la casa 3 (Ca3)

Construir el **grafo de planificación relajado** (es decir, sin tener en cuenta los efectos negativos de las acciones y sin calcular las relaciones de exclusión mutua), y contestar a las siguientes preguntas:

- 1) Calcula el valor de las heurísticas h_sum y h_max para los dos objetivos definidos.
- 2) Extrae un plan relajado para los dos objetivos sobre el grafo de planificación relajado para los dos objetivos dados. Mostrar la extracción del plan relajado y demostrar cómo este plan se puede extraer en tiempo polinómico y sin necesidad de operaciones de backtracking.
- 3) ¿Cuál de las tres heurísticas calculadas (h_sum , h_max , $plan_relajado$) es la más informada para este problema? ¿Por qué?