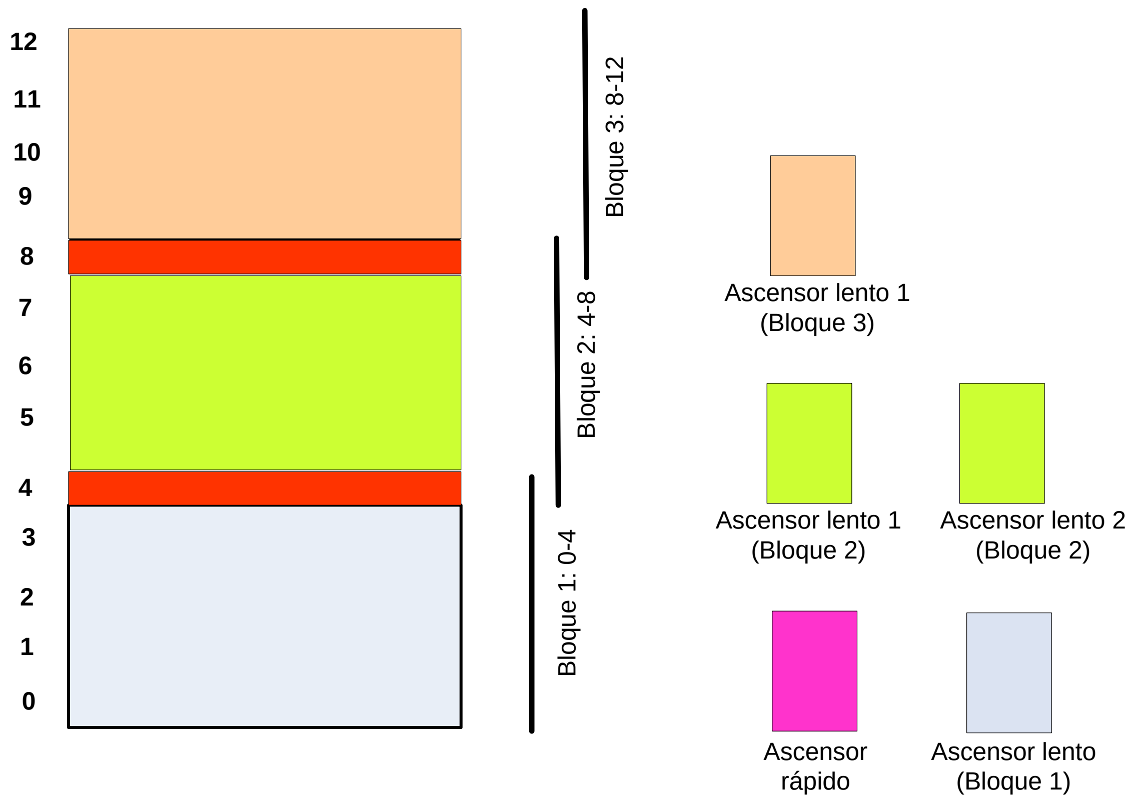
Dominio Ascensores

Ejercicio 1: Dominio proposicional

Se dispone de un edificio de 13 plantas (numeradas del 0 al 12) formado por tres bloques. El bloque 1 va de la planta 0 a la planta 4; el bloque 2 va de la planta 4 a la planta 8; el bloque 3 va de la planta 8 a la planta 12. Los bloques 1 y 2 comparten la planta 4, y los bloques 2 y 3 comparten la planta 8.

En el edifico hay un ascensor rápido que solo para en las plantas pares (0,2,4,6,8,10,12). Además, cada bloque dispone de uno o dos ascensores lentos, tal y como se muestra en la figura, que solo se pueden mover entre las plantas del bloque correspondiente.



El ascensor rápido tiene una capacidad para tres personas, y los ascensores lentos tienen capacidad para dos personas.

La situación inicial es la siguiente. Hay 5 pasajeros que están situados en las siguientes plantas:

1) Pasajero 0 está en la planta 2
2) Pasajero 1 está en la planta 4
3) Pasajero 2 está en la planta 1
4) Pasajero 3 está en la planta 8
5) Pasajero 4 está en la planta 1

El objetivo es llevar a los pasajeros a las siguientes plantas:

1) Pasajero 0 a la planta 3
2) Pasajero 1 a la planta 11
3) Pasajero 2 a la planta 12
4) Pasajero 3 a la planta 1
5) Pasajero 4 a la planta 9

Se pide:

a) Definir el dominio correspondiente con las posibles acciones u operadores a aplicar utilizando el lenguaje PDDL.
b) Definir la instancia del problema, describiendo los predicados que definen la situación inicial y el objetivo del problema.
c) 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 cambiando la situación inicial y final. Se pueden modificar los siguientes datos:

- Cambiar plantas de origen y destino de los pasajeros
- Incluir nuevos pasajeros
- Cambiar la capacidad de los ascensores
- Incluir nuevos ascensores, rápidos o lentos. El ascensor rápido puede moverse por plantas distintas a las pares (por ejemplo, múltiplos de 3) pero nunca por todas las plantas del edificio. Los ascensores lentos siempre deben moverse dentro de las plantas del bloque.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIIC
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Ejercicio 2: Dominio temporal

Ascensores lentos:

- Subir/bajar una planta: 12 unidades de tiempo
- Subir/bajar dos plantas: 20 u.t.
- Subir/bajar tres plantas: 28 u.t.
- Subir/bajar cuatro plantas: 36 u.t.

Ascensor rápido:

- subir dos plantas: 11 u.t.
- subir cuatro plantas: 13 u.t.
- subir seis plantas: 15 u.t.
- subir ocho subir plantas: 17 u.t.
- subir 10 plantas: 19 u.t.
- subir 12 plantas: 21 u.t.
- bajar dos plantas: 10 u.t.
- bajar cuatro plantas: 12 u.t.
- bajar seis plantas: 14 u.t.
- bajar ocho subir plantas: 16 u.t.
- bajar 10 plantas: 18 u.t.
- bajar 12 plantas: 20 u.t.

Personas:

- subir/bajar del ascensor: 2 u.t.

1) Definir el dominio temporal. Indicar si habéis realizado algún cambio respecto a la codificación del dominio proposicional.
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, y considerando únicamente el objetivo (Pasajero 2 a la planta 12), 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.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Ejercicio 5: Graphplan (sin tiempo ni recursos)

Partiendo del problema del primer apartado, y dado solo los dos primeros objetivos:

a) Pasajero 0 a la planta 3
b) Pasajero 1 a la planta 11

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?