Laboratorio: Planificación en STRIPS/PDDL

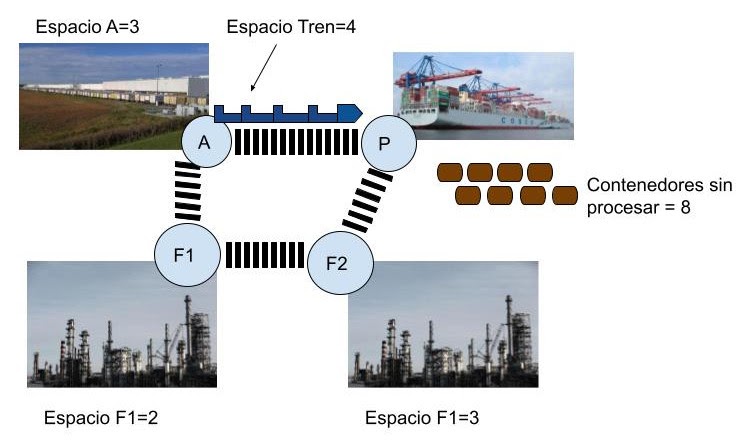
## **Objetivos de la actividad**

Con esta actividad se pretende incentivar la resolución autónoma de un caso que requiere acceso a los planificadores del estado del arte para solucionar un problema en STRIPS/PDDL. Para ello es preciso ser capaz de representar el problema en el lenguaje PDDL, generar planes y comparar resultados.

## **Descripción de la actividad**

En una ciudad existen las siguientes tipos de localización: puerto (P), almacén (A) y fábrica (F1 y F2). En el puerto hay contenedores que hay que procesar. En las fábricas se procesan los contenedores. Una vez procesados, los contenedores deben llevarse al almacén. Desde el almacén, los contenedores ya procesados desaparecen del sistema. Las distintas ubicaciones tienen una capacidad limitada (excepto el puerto). Es posible llevarse al almacén los contenedores de material sin procesar, para usarlo como almacenamiento temporal.

La siguiente figura que describe una posible situación inicial (problema) con dos fábricas, que es el caso básico que hay que resolver.



Fuente de las imágenes: free\_images.com

Las localizaciones están conectadas por la vía del tren en un anillo como se ve en la figura (A-P-F2-F1-A). Hay un único tren, que puede moverse por la vía en ambas direcciones, y cargar y descargar contenedores, siempre que no exceda su capacidad al cargar, ni la de la ubicación donde descarga.

El tren puede realizar las siguientes acciones:

* Quedarse **parado**.
* **Mover** a una localización **directamente conectada** sobre la vía (es decir, de A a P, o de F1 a A, etc.). **Puede moverse en direcciones opuestas en acciones sucesivas** (es decir, puede darse la vuelta ya que tiene máquina en ambos extremos).
* **Cargar** un contenedor en cualquier localización donde se encuentre, **siempre y cuando no se exceda el espacio disponible**.
* **Descargar** un contenedor en cualquier localización donde se encuentre, a excepción del puerto. Recuerde que al descargar no se puede exceder la capacidad de la localización.

El procesamiento de los contenedores está abstraído y es automático. Es decir, no no son acciones aparte:

* Todos los contenedores están inicialmente sin procesar
* Cualquier contenedor sin procesar que esté en una fábrica pasa a estar "procesado" cuando el tren realiza la **siguiente** acción. Nota: no es preciso considerar explícitamente el tiempo, basta con añadir este consecuente a las acciones del tren.
* Cualquier contenedor procesado que sea descargado en el almacén **desaparece** del mismo (se lo lleva una flota de camiones).

El estado final consiste en que **todos los contenedores desaparezcan** (es decir, se procesen y luego sean llevados al almacén, donde salen del sistema).

# **Parte 1**

Se deberá modelar en PDDL (dominio.pddl y problema.pddl) el dominio del problema y el escenario particular que se plantea en la situación inicial, y resolver el problema con algún planificador del estado del arte (basta uno).

Si se usa el plugin de PDDL de Visual Studio Code[[1]](#footnote-2), se pueden utilizar varios planificadores de forma nativa[[2]](#footnote-3) y bastante cómoda. También se admite que el estudiante busque y configure por su cuenta los planificadores. En las siguientes direcciones se puede descargar el código de planificadores de uso frecuente:

* LPG-TD<https://lpg.unibs.it/lpg/download-lpg-td.html> (windows, linux)
* FF<https://fai.cs.uni-saarland.de/hoffmann/ff/FF-v2.3.tgz> (Linux).

Si lo desea, se puede realizar la prueba con más de un planificador y comparar los resultados. Debe documentarse qué planificador se ha usado.

# **Parte 2**

Adicionalmente, se deben modelar dos situaciones (dos problemas) diferentes a la planteada en esta actividad: por ejemplo, cambiar las conexiones entre las localizaciones, añadir una vía entre A y F2, cambiar las capacidades de cada elemento, o el número de contenedores de partida, etc. Y se deben resolver los 2 problemas con el mismo planificador empleado en la parte 1.

## **Documentos a entregar**

* Códigos fuente de los ficheros (dominio.pddl y problema.pddl)
* Memoria explicativa en Word o PDF de no más de 20 páginas en la que se incluya:
  + Informe y documentación del código del dominio y código del problema, de forma que pueda entenderse la implementación del alumno.
  + Capturas de pantalla de la salida de la ejecución del planificador
  + Resultados de la situación básica (de la figura) y las dos situaciones adicionales. Deben incluirse tiempos de ejecución, plan obtenido, si el plan es óptimo o no, etc.
  + Incluir las dificultades encontradas.
  + Referencias con Normas APA. (se penaliza si no tienen Normas APA)

## **Aspectos a tener en cuenta**

* La actividad es individual. Aunque durante la sesión de laboratorio la desarrollamos en grupos.
* El código PDDL debe ser de la autoría del o de la estudiante.
* Cualquier página, libro, u otro material consultado debe ser referenciado con Normas APA. Tenga en cuenta que no referenciar correctamente los materiales será considerado plagio y por lo tanto afectará a la nota.

**Rúbrica**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | Presenta correctamente los códigos del dominio y los problemas en STRIPS/PDDL | 4 | 40% |
| Criterio 2 | Se resuelve correctamente el problema con un planificador del estado del arte, y se comentan adecuadamente los resultados obtenidos. | 2 | 20% |
| Criterio 3 | Modela dos situaciones diferentes a la planteada y las resuelve con 1 planificador. Se analiza el tiempo necesario para la resolución, solución obtenida, etc. | 2 | 20% |
| Criterio 4 | Entrega un documento final de calidad con comparaciones entre los casos (y planificadores), razonando correctamente las diferencias encontradas. Se presentan referencias a casos reales similares. | 2 | 20% |
|  |  | **10** | **100 %** |

1. <https://marketplace.visualstudio.com/items?itemName=jan-dolejsi.pddl> [↑](#footnote-ref-2)
2. <https://github.com/jan-dolejsi/vscode-pddl/wiki/Configuring-the-PDDL-planner>. [↑](#footnote-ref-3)