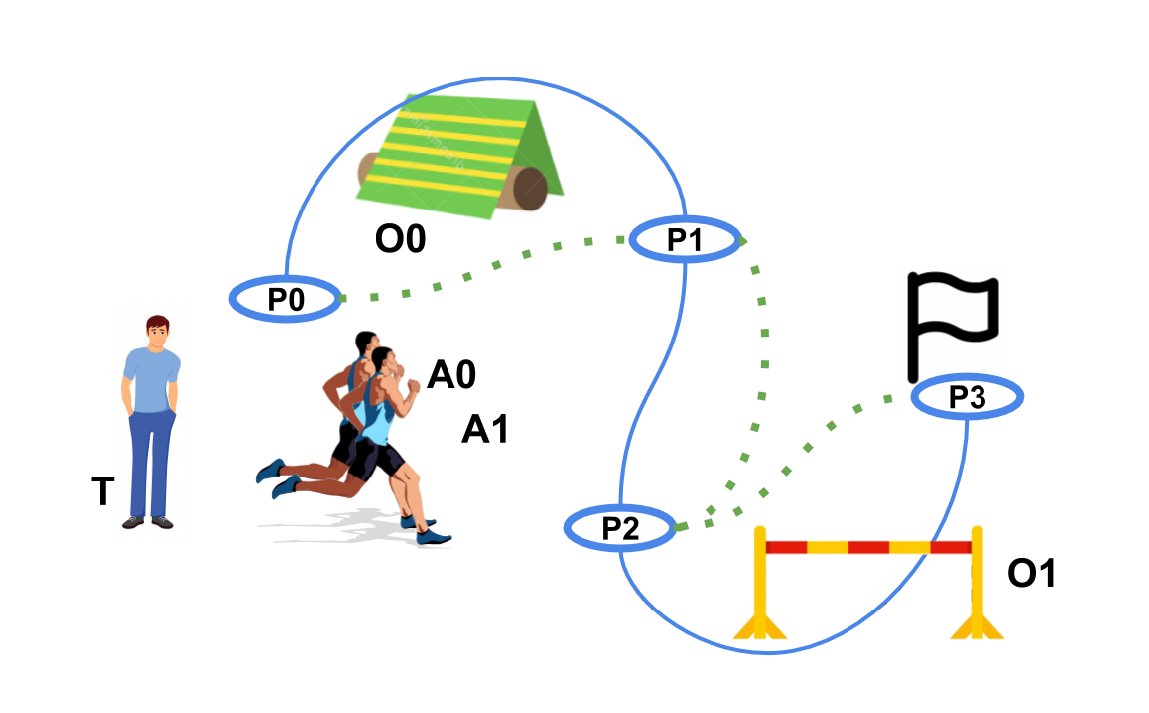
**PLANIFICACIÓN AUTOMÁTICA**

**PL3 - Integrate path-planning and task-planning**

**Grado de Ingeniería Informática – Curso 2019/2020**

****

*Guillermo Perales Gallar*

*Iván Cantero Martín*

**Fallo compilación del comando:**

**python r2p2.py --scenario ../conf/scenario-planning.json**

A la hora de integrar el controlador con el Rover desarrollado en PDDL hemos detectado el error comentado por la profesora. Hemos implementado el algoritmo theta\* y las heurísticas de forma correcta pero ocurre un error en la integración y no se ejecuta correctamente el camino del robot. Este error es ajeno a nosotros tal y como nos comentó la profesora y hemos intentado corregirlo modificando el archivo planning.txt de diferentes maneras, incluso con el archivo que hay en el GitHub de la práctica pero sin solución satisfactoria.

Lo que hemos realizado en esta práctica se podría denominar como la integración entre path-planning y los planificadores PDDL.

Hay que tener en cuenta que estamos planificando y que una vez tenemos un plan debemos **ejecutarlo**.

Esto es lo que falta, una **Arquitectura de Control** la cual se adapte a cada situación concreta, y si tiene que replanificar que lo haga y ejecute el nuevo plan.

Más concretamente, seleccionar un ejecutor y utilizarlo, ya que podría darse el caso de tener que realizar una replanificación si la ejecución no sale como se esperaba con el plan inicial.

Ya conocíamos que PDDL puede crear planes en referencia a un dominio y un plan concretos, pero tiene dificultades a la hora de obtener el camino óptimo y para eso se utiliza path-planning, haciendo uso de diversos algoritmos (Dijkstra, A\* y Theta\*) y heurísticas (Euclidean, Manhattan y Octile).

Hemos observado mediante experimentación cómo, dependiendo tanto del algoritmo como de la heurística que usemos, el resultado a la hora de encontrar el camino cambia, siendo unas opciones mejores en algún caso y peores en otros, es decir, que no hay una solución estándar óptima para todos los casos.

Se puede observar en las imágenes en la carpeta adjunta *r2p2 > Imagenes 30 grid inicio 4,4 y final 26,26*, cómo se realizan todos los algoritmos con todas las heurísticas posibles. En estas imágenes aparece también el resultado de la ejecución, donde se puede observar el coste del camino y el total de nodos expandidos, entre otra información relevante con respecto al algoritmo en cuestión.

Por normal general, la heurística Octile es la que mejor resultados nos ha dado.

Estas experimentaciones, aunque no han requerido mucho de nuestra parte, en problemas de mayor magnitud es posible que requieran demasiado tiempo.

**Por ello, determinamos que lo que faltaría por hacer en un fututo es, una vez se platee un problema, automatizar el sistema y que sea capaz de elegir qué algoritmo y con qué heurística trabajar en cada momento, siendo capaz de adaptarse y aprender según va solucionando más escenarios.**