

PROBLEMA 3 SID

Decisiones de diseño

La ontología de esta práctica la hemos diseñado en formato TURTLE, ya que al guardarlo con el formato OWL, no podíamos cargarlo en el código del agente.

Respecto a sus funcionalidades, permite expresar lo pedido en el enunciado del problema. Tenemos dos clases principales: *Nodo* y *Agente*. *Nodo* representa a los distintos nodos del mapa (que en este problema son 5). Para representar los roles que pueden tener los agentes en el entorno, hemos creado 3 subclases de *Agente*: *Explorador*, *Almacenamiento* y *Recolector*.

Para mantener la consistencia del mapa, hemos creado la relación *es_contiguo_a*, que relaciona a dos *Nodos* si son contiguos. Esta relación es irreflexiva, ya que un nodo no puede ser contiguo a sí mismo y es simétrica, es decir, si un *Nodo A* es contiguo a un *Nodo B*, entonces el *Nodo B* también es contiguo al *Nodo A*.

Con tal de relacionar la posición de cada agente en el mapa, hemos creado dos relaciones: *esta_en* y *tiene_agente*. La primera de estas relaciones, liga un *Agente* con un *Nodo*, es decir, nos dice en qué *Nodo* se encuentra. La segunda relación, *tiene_agente*, relaciona un *Nodo* con un *Agente* de forma similar a la primera relación descrita. Por razones obvias, estas dos relaciones son las inversas el uno del otro.

En referencia al cómputo de la distancia a un nodo *Almacenamiento*, hemos decidido hacerlo mediante clases equivalentes. Primero hemos creado una clase equivalente *NodoAlmacenador*, que es un *Nodo* que tiene algún *Agente Almacenamiento*. Entonces, para los nodos que se encuentran a distancia 1, hemos creado la clase equivalente *NodoDistancia1*, que tiene como condición necesaria tener estar contiguo a un *NodoAlmacenador*. Hemos hecho lo mismo para los nodos que se encuentran a distancia 2, es decir, una clase equivalente *NodoDistancia2*, pero en este caso tiene que estar contiguo a un *NodoDistancia1*. Al tener el mapa solo 5 nodos, la máxima distancia es 2, por lo tanto, hemos decidido crear solo estas dos clases equivalentes.

Finalmente hemos añadido una clase equivalente *RecolectorDescarga*, que nos permitirá inferir a los agentes recolectores cuando se encuentren en un nodo contiguo a un almacenador, o *NodoDistancia1*. De esta manera, un agente recolector sabrá si puede descargar los minerales que lleve. Esta inferencia que comentamos, no fue algo trivial de decidir, ya que al principio no sabíamos cómo hacer que un agente supiera que sí podía almacenar lo que llevaba. Al final, crear una clase equivalente, nos pareció la mejor solución ya que, además de ser claramente representativa, permite una clara y sencilla inferencia por parte del razonador.

En esta práctica, el agente *Player* lo hemos diseñado como un agente racional con la capacidad de deducir si su posición pertenece en un nodo de descarga o no. Respecto al

camino a seguir para poder almacenar, hemos optado por que el agente siga un camino predefinido tal que Nodo5 -> Nodo1. Este último nodo es contiguo al nodo que contiene el almacenador y, por tanto, es un nodo objetivo para este entregable.

En referencia al estado interno del agente, hemos decidido crear una única *capability*, que engloba todas las creencias que definen el sistema en el que se encuentra el agente. Dentro de su base de creencias, o *BeliefBase*, hemos creado un conjunto *beliefs* para poder encapsular los siguientes conceptos:

- Rol del agente
- Posición del agente
- Modelo Jena

Este último *belief*, será el que nos permitirá poder ir actualizando la posición del agente, saber donde está, si puede almacenar los minerales...

Extra

Para el apartado extra de la práctica, nuestra ontología incluye las clases, relaciones y funciones necesarias para inferir la distancia a la que se encuentra cada agente recolector de un agente almacenamiento, pero no es capaz de identificar la mínima, es decir, infiere todas las distancias que les permite llegar a un agente almacenamiento.

Reparto del trabajo

La idea inicial y toma de decisiones respecto a la Ontología y el funcionamiento y comportamiento del agente lo realizamos entre los 4 miembros del equipo en conjunto. La implementación de dichas ideas las dividimos en 2 tareas distintas.

Diseño e implementación de la Ontología:

- Victor Teixidó López
- Jiabo Wang

Diseño e implementación del Agente:

- Jia Long Ji Qiu
- You Wu