

# Problema 3 de 4

## SID 2122q2

Abril/Mayo 2022

En el entorno de la práctica de laboratorio<sup>1</sup> (Dedale) el mapa es un grafo conexo, con presencia de recursos en algunos nodos, y por el que se pueden mover los agentes (ver Figura 1).

Los agentes pueden tener diversos roles, incluyendo recolector y almacenamiento. Los agentes recolectores pueden recoger recursos y transportarlos, pero generalmente tienen una capacidad de carga limitada. Los agentes almacenamiento no pueden recoger recursos, pero suelen tener una capacidad de carga muy amplia.

Para una recogida eficiente de recursos, es necesario que los agentes de ambos roles cooperen para poder optimizar el uso de la capacidad de carga colectiva. Para ello, los agentes recolectores tienen disponible una acción *EmptyMyBackPack(destinationAgent)* que les permite transferir su carga a un agente almacenamiento que se encuentre en un nodo contiguo al suyo.

## 1 Enunciado

Usando la plataforma de JADE, las librerías de JADE, BDI4JADE y Jena:

- Diseñar una ontología en formato OWL que permita expresar:
  - La topología del mapa.
  - La existencia de agentes de diferentes roles: explorador, recolector, almacenamiento.
  - La posición de cada agente en el mapa.

Esta ontología debería realizar las siguientes inferencias:

- Un nodo está a distancia 1 de (es contiguo con) al menos un nodo en el que se encuentra un agente almacenamiento.
- Un agente recolector puede realizar una descarga en el nodo en el que está (porque hay un agente almacenamiento en un nodo contiguo al suyo).

---

<sup>1</sup>Ver enunciado de la práctica para entender el contexto de este problema.

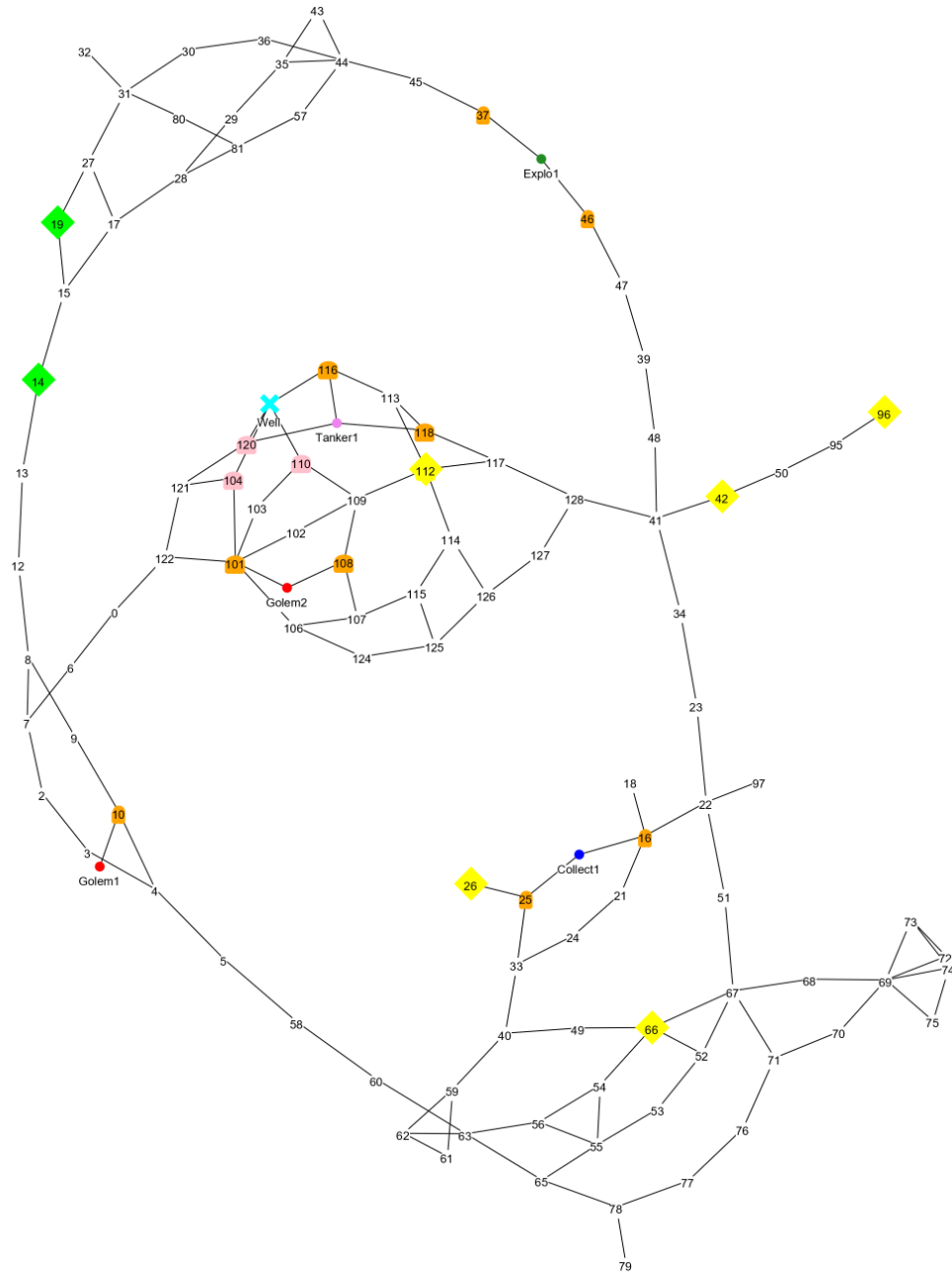


Figure 1: Ejemplo de mapa

- (Opcional) Un nodo está a distancia N del nodo más cercano en el que se encuentra un agente almacenamiento<sup>2</sup>.
- Cread un agente subclase de *SingleCapabilityAgent* de BDI4JADE cuyos *beliefs* sean:
  - El grafo correspondiente al mapa del entorno es el de la Figura 2.
  - El agente tiene el rol recolector y empieza en el nodo 5.
  - Hay un agente almacenamiento en el nodo 2.
  - Un objeto *Model* de Jena cargado con la ontología que habéis creado.

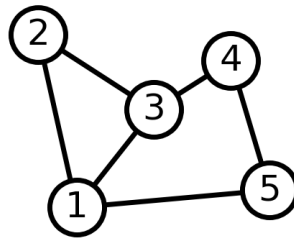


Figure 2: Mapa para el problema

El objetivo del agente es llegar a un nodo que permita descargar recursos (es decir, contiguo al nodo 2 en este caso). El cumplimiento de este objetivo deberá ser validado por la existencia de un belief que indique que dicha condición (estar en condiciones de descargar recursos) se da. **Este belief deberá ser una inferencia producida por la ontología que habréis creado, utilizando el modelo de Jena.** Para que el agente tenga acceso a esta inferencia será necesario asignar, a la *capability* de vuestro agente, una función de revisión de beliefs<sup>3</sup> creada por vosotros cuya implementación consista en:

1. Actualizar el modelo de la ontología con las instancias y propiedades que representen el estado interno actual del agente.
2. Una vez actualizada la ontología, obtener de ella (usando la Ontology API o la SPARQL API de Jena) las inferencias relevantes (e.g. el agente recolector puede realizar la descarga) y actualizar los beliefs en base a estas inferencias.

<sup>2</sup>Lo podéis hacer para un caso general o bien para un rango acotado de N, como por ejemplo de 1 a 10.

<sup>3</sup>Usando el método `Capability.setBeliefRevisionStrategy`: <https://www.inf.ufrgs.br/prosoft/bdi4jade/api/bdi4jade/core/Capability.html#setBeliefRevisionStrategy-bdi4jade.reasoning.BeliefRevisionStrategy->

No se requiere que el agente corra en el entorno Dedale ni que se comunique con un *AbstractDedaleAgent*. Del mismo modo, no hace falta que uséis un algoritmo de pathfinding para “desplazar” el agente en el grafo, lo podéis hacer en base a un camino prediseñado (5-1 o 5-4-3, por ejemplo). El PlanBody, en cualquier caso, deberá estar implementado de manera que no impida al agente revisar sus beliefs cada vez que se haya movido de un nodo a otro.

## 2 Entrega

En la entrega tenéis que incluir:

- Un documento breve (.txt o .pdf) que explique:
  - Las decisiones que habéis tomado para implementar la ontología y el agente.
  - Cómo os habéis repartido las tareas entre los autores de la entrega.
- Un fichero .owl con la ontología.
- El código fuente del agente.

## 3 Evaluación

La evaluación se dividirá en los siguientes aspectos:

- La ontología cubre todos los aspectos mencionados en el enunciado, de manera implícita (a través de inferencias) o de manera explícita. La ontología contiene anotaciones que ayudan a entender las clases, relaciones y funciones definidas. (3/10 puntos)
- Todas las clases son completas y coherentes en cuanto a sus axiomas, incluyendo condiciones necesarias y suficientes. (2/10 puntos)
- Todas las relaciones y funciones definidas son completas en cuanto a sus propiedades (características, dominio y rango). (2/10 puntos)
- El comportamiento del agente es correcto y el diseño de sus objetivos y planes es coherente con respecto a este comportamiento. (1/10 puntos)
- La función de revisión de beliefs utiliza la ontología como un modelo de Jena, actualizándola en cada iteración del ciclo BDI en base al estado del agente y obteniendo las inferencias de manera que ayuden en la consecución de sus objetivos. (2/10 puntos)

- *Extra:* La ontología incluye las clases, relaciones y funciones necesarias para poder inferir la distancia mínima a la que se encuentra cada agente recolector de un agente almacenamiento. Para simplificar, podéis asumir que la distancia siempre será en cualquier caso menor que 10. (2/10 puntos)

La entrega y corrección del problema se hará en grupos de como máximo 3 personas y contará un 1/4 de la nota de problemas. Deberéis entregar vuestra solución antes del final del viernes 20 de mayo en el espacio que se habilitará en el Racó.