

Problema 2 de 4

SID 2223q2

Abril/Mayo 2023

En el entorno de la práctica de laboratorio¹ (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 Protégé y la librería Jena:

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

Esta ontología debería realizar las siguientes inferencias (usando *clases equivalentes*):

- Un nodo está a distancia 1 de (es contiguo con) al menos un nodo en el que se encuentra un agente almacenamiento.

¹Ver enunciado de la práctica para entender el contexto de este problema.

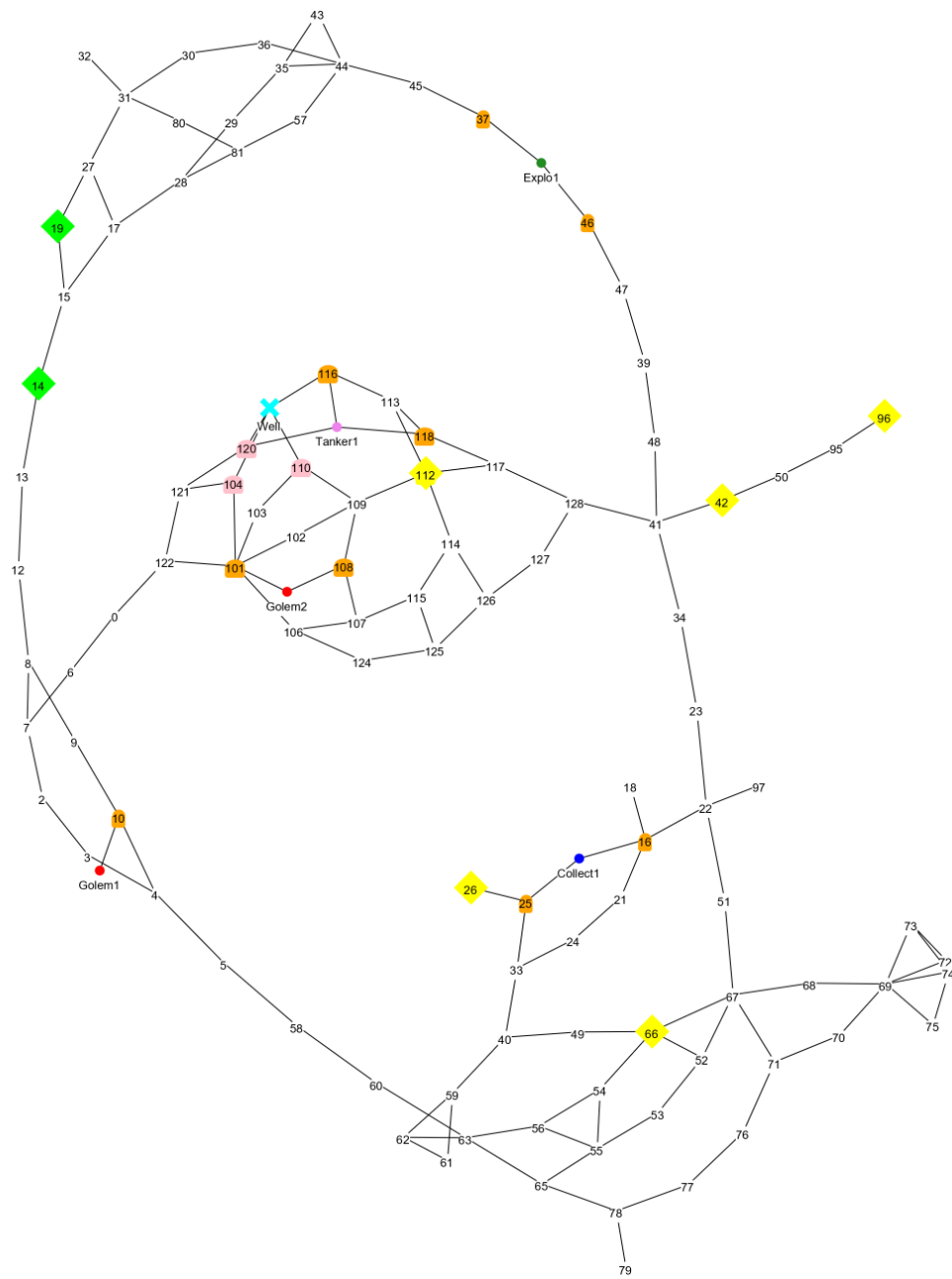


Figure 1: Ejemplo de mapa

- 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).
 - Implementad un *behaviour* compuesto para un agente Dedale que implemente el siguiente comportamiento:
 - El agente carga la ontología que habéis creado en un *Model* de Jena.
 - Tras la carga de la ontología y tras cada movimiento del agente, se crean o modifican las instancias (*Individuals* de Jena) necesarias para que la ontología represente:
 - * El grafo correspondiente al mapa del entorno conocido por el agente.
 - * El propio agente, su rol y su posición.
 - (*Optional:*) Tras cada mensaje recibido por otro agente, se crean o modifican las instancias necesarias para que la ontología represente:
 - * Los otros agentes, sus roles y posiciones.
- Os podéis inventar el formato y contenido de los mensajes que vuestros agentes se intercambian. Podéis usar la entrega 1 como base para esta parte opcional.

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 *behaviour*.
 - 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 de la ontología 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). (1/10 puntos)
- El comportamiento del agente se ajusta al enunciado y el diseño de *behaviour* y *subbehaviours* permite un funcionamiento correcto combinado con otros *behaviours* que pueda tener el agente de manera concurrente. (4/10 puntos)
- (*Extra:*) El comportamiento del *behaviour* permite actualizar la ontología en base a los mensajes recibidos de los otros agentes. (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 jueves 4 de mayo en el espacio que se habilitará en el Racó.