

Práctica 3

Técnicas de los Sistemas Inteligentes

Carmen Azorín Martí

4 de junio de 2025

Introducción

En esta práctica se resuelven varios ejercicios relacionados con los contenidos de la asignatura Técnicas de los Sistemas Inteligentes.

1. Ejercicio 1

La idea del primer ejercicio es modelar el mundo de forma que los personajes puedan desplazarse por él y extraer recursos. El objetivo es generar madera en la localización de Fangorn.

1.1. El dominio

El dominio tiene los tipos: personajes (que solamente incluye Enanos y Hobbits como constantes), recurso (incluye Madera, Mineral, Mithril... como constantes) y localización.

Y los predicados permiten representar lo siguiente: Los predicados definidos son: **en** y **enRecurso** (localización de personajes y recursos), **conectado** (conexión entre dos localizaciones), **esRol** (rol de un personaje: Enano o Hobbit), **disponible** (disponibilidad del personaje), **extrayendo** (un personaje está extrayendo un recurso), **generando** (el recurso se está generando), y **esCategoria** (categoría de un recurso: Mineral, Mithril, Alimento...).

Y las acciones son las siguientes:

- **Viajar**: permite a un personaje moverse entre dos localizaciones conectadas. Para ello, el personaje debe estar en la primera localización y que ésta esté conectada con la segunda localización. EL efecto es que el personaje deja de estar en la primera localización y pasa a estar en la segunda.
- **ExtraerRecurso**: permite a un Enano disponible extraer un recurso de una localización. Para ello, el personaje debe ser un Enano, estar disponible y que tanto él como el recurso se encuentren en la localización. El efecto es que se activa la extracción y se considera que el recurso se está generando.

1.2. Problema

El problema declara los objetos de los personajes (Enano1, Enano2, Enano3), los 5 recursos según su localización (r_fangorn, r_moria...) y las 20 localizaciones (Fangorn, Moria...).

El estado inicial establece que Enano1 comienza en Tharbad, mientras que Enano2 y Hobbit1 se encuentran en Isengard. Enano1 y Hobbit1 están disponibles, pero Enano2 no lo está. Los recursos están correctamente categorizados y ubicados (por ejemplo, r_fangorn se encuentra en Fangorn y es de tipo Madera). Además, la conectividad entre localizaciones está definida de forma bidireccional, lo cual ha sido generado automáticamente por ChatGPT.

El objetivo será que se esté generando el recurso r_fangorn, es decir, que haya un personaje en Fangorn extrayendo Madera.

1.3. Resultado

```
viajar enano1 tharbad helmsdeep (1)
viajar enano1 helmsdeep isengard (1)
viajar enano1 isengard fangorn (1)
extraerrecurso enano1 r_fangorn fangorn (1)
```

2. Ejercicio 2

En este ejercicio se busca reflejar el viaje de la Comunidad del Anillo para destruir el Anillo Único de Orodruin. Para ello, se necesita formar una comunidad formada por un Hobbit y un Mago, recoger los objetos (Anillo, Mithril y Espada) y llegar al lugar de destrucción.

2.1. Dominio

Este dominio extiende al anterior. Los tipos son los mismos, pero se añade una nueva constante de personaje que es Mago.

Los predicados nuevos son los siguientes: **esAnillo** y **esEspada**, que indican si un recurso es el Anillo o la Espada (ya que no son categorías); **miembroComunidad** y **comunidadFormada**, que indican si un personaje pertenece a la comunidad y si esta ya ha sido creada; **posee**, que refleja si un personaje guarda uno de los recursos clave; **portador**, que identifica al personaje que porta el Anillo y recogerá los objetos necesarios; **lugarDestruccion**, que indica dónde debe destruirse el Anillo; y **anilloDestruido**, que señala si ya se ha cumplido el objetivo final del dominio.

Las acciones posibles son:

- **formarComunidad**: forma la Comunidad. Para ello, los integrantes deben ser un Hobbit y un Mago, que se encuentren en la misma localización y que la Comunidad no esté ya formada. El efecto será la formación de la Comunidad y los personajes siendo miembros.
- **viajarComunidad**: todos los miembros de la Comunidad pasan de estar en una localización a estar en otra conectada. El efecto será que dejan de estar en la primera localización y están en la otra.

- **Viajar:** se modifica la acción viajar del ejercicio anterior. Para que un personaje viaje, debe estar fuera de la Comunidad, o bien que esta Comunidad no esté formada.
- **recogerObjeto:** para que un personaje pueda recoger un recurso, la Comunidad debe estar formada y debe ser miembro de ésta. Además, debe estar en la misma localización que el recurso. Si es el Anillo, es personaje debe ser Hobbit; si es Mithril o Espada, debe recogerlo el poseedor del anillo, es decir, el Hobbit. El efecto será que el personaje posee el recurso y será el portador.
- **destruirAnillo:** el portador destruye el anillo en la localización de destrucción cuando posea los 3 recursos necesarios.

2.2. Problema

Para el problema se incluyen los personajes: Hobbit4, Mago1-2 y Enano1, que están distribuidos en distintas localizaciones. Mago1 comienza en Rivendell, donde también se encuentra el Anillo. El Mithril está en Moria y la Espada en Lothlorien. Finalmente, el lugar de destrucción del Anillo es Orodruin. Los objetos clave como Anillo y Espada son recursos.

El objetivo será que el anillo quede destruido.

2.3. Resultado

```
viajar hobbit4 bree rivendell (1)
formarcomunidad hobbit4 mago1 rivendell (1)
recogerobjeto hobbit4 anillo rivendell (1)
viajarcomunidad rivendell moria hobbit4 mago1 (1)
recogerobjeto hobbit4 r_moria_mithril moria (1)
viajarcomunidad moria lothlorien hobbit4 mago1 (1)
recogerobjeto hobbit4 espada lothlorien (1)
viajarcomunidad lothlorien amonhen hobbit4 mago1 (1)
viajarcomunidad amonhen deadmarshes hobbit4 mago1 (1)
viajarcomunidad deadmarshes minasmorgul hobbit4 mago1 (1)
viajarcomunidad minasmorgul orodruin hobbit4 mago1 (1)
destruiranillo hobbit4 orodruin (1)
```

3. Ejercicio 3

Tamaño de la Comunidad	Tiempo de Planificación	Longitud del Plan
1 hobbit y 1 mago	0.78s	12 step(s)
2 hobbits y 1 mago	4.60s	14 step(s)
3 hobbits y 1 mago	28.73s	15 step(s)
3 hobbits, 1 mago y 1 elfo	727.84s	18 step(s)

Cuadro 1: Resultados según el tamaño de la Comunidad del Anillo

Para reducir el tiempo de las acciones de `viajarComunidad`, se ha decidido tener condiciones redundantes como que: dos personajes no pueden coincidir, y qué roles tiene

cada personaje, para que no tenga tantos entre los que buscar (por ejemplo, obligar a que un parámetro sea hobbit o mago). Estas condiciones no cambian el significado del dominio, pero ayudan a reducir las posibles combinaciones del planificador.

A medida que se incluyen acciones, objetos y nodos en el mapa, los tiempos de planificación escalan de forma exponencial. Esto se debe a que cada nuevo elemento aumenta el tamaño del espacio de búsqueda de manera combinatoria. Más acciones implican más ramas posibles desde cada estado. Más objetos (como personajes o recursos) multiplican las posibles especificaciones de cada acción. Por ejemplo, si la acción **formarComunidad** toma 4 personajes, en vez de 2, el número de posibles subconjuntos válidos se multiplica drásticamente. Más nodos en el mapa implican más rutas posibles y, en consecuencia, más alternativas por las que desplazarse o encontrarse. Si se añade una localización más en el primer apartado, el Hobbit tiene una localización más en la que encontrarse con el Mago y se multiplican las posibles rutas, ya que ahora se puede pasar por la nueva ciudad. Y este es el caso más simple.

Si la comunidad estuviese formada por 3 hobbits, 1 mago, 1 elfo, 1 enano y 2 humanos, el ordenador no podría encontrar el plan sin apagarse o interrumpir la ejecución. Podríamos estimar horas o días. Aunque sería posible añadir más restricciones o precondiciones redundantes para reducir el espacio de búsqueda, al final el problema sigue siendo el mismo: cuantas más cosas hay que tener en cuenta, más difícil se vuelve encontrar un plan. El número de combinaciones posibles crece tan rápido que llega un punto que es prácticamente imposible planificar algo sin que se bloquee el ordenador.

4. Ejercicio 4

Este ejercicio extiende el ejercicio 2 introduciendo construcción de edificios. El objetivo es que construya una fortaleza en la localización de Isengard, utilizando recursos extraídos por distintos personajes según su rol.

4.1. Dominio

Se incorpora el tipo edificio, con dos valores posibles: Extractor y Fortaleza. Aparecen dos nuevos predicados que indican el tipo de Edificio y dónde se encuentra un edificio construido. Y aparecen dos nuevas acciones:

- **ExtraerRecurso**: un personaje disponible y en la localización de un recurso puede extraerlo bajo ciertas condiciones. La madera solo puede ser extraída por un Enano, el mineral solo puede extraerse por un Humano, El Mithril solo puede extraerse por un Enano y si ya hay un Extractor construido en la misma localización. Además, todos los Hobbit deben estar generando Alimento antes de que se puedan extraer otros recursos.
- **Construir**: un Extractor solo puede construirse por un Humano y si se esta generando mineral. Una fortaleza tiene que construirla un Mago siempre que se este generando tanto madera como mithril.

4.2. Problema

El estado inicial incluye: 4 Hobbits, 2 Magos, 2 Enanos, 1 Elfo y 2 Humanos. Los recursos y localizaciones de los ejercicios anterior. Y ahora todos los personajes comienzan

disponibles y se definen las construcciones Edificio y Fortaleza.

El objetivo es que se haya construido una Fortaleza en Isengard.

4.3. Resultado

```
viajar hobbit4 bree hobbiton (1)
viajar enano1 fangorn amonhen (1)
viajar enano1 amonhen lothlorien (1)
viajar enano1 lothlorien moria (1)
extraerrecurso hobbit1 hobbiton alimento (1)
extraerrecurso hobbit2 hobbiton alimento (1)
extraerrecurso hobbit3 hobbiton alimento (1)
extraerrecurso hobbit4 hobbiton alimento (1)
viajar humano2 edoras helmsdeep (1)
viajar humano2 helmsdeep tharbad (1)
viajar humano2 tharbad bree (1)
viajar humano2 bree rivendell (1)
viajar humano2 rivendell moria (1)
viajar enano2 erebor mirkwood (1)
extraerrecurso enano2 mirkwood r_mirkwood (1)
extraerrecurso humano2 moria r_moria_mineral (1)
viajar humano1 edoras helmsdeep (1)
viajar humano1 helmsdeep tharbad (1)
viajar humano1 tharbad bree (1)
viajar humano1 bree rivendell (1)
viajar humano1 rivendell moria (1)
construir humano1 extractor moria (1)
extraerrecurso enano1 moria r_moria_mithril (1)
```

5. Ejercicio 5

Para implementar este ejercicio he usado chatGPT para que me guíe en cómo desarrollarlo. Lo que finalmente he hecho ha sido usar `(:requirements :action-costs)` y definir una función `total-cost` para medir el coste total del plan, y `costeCamino ?l1 ?l2` para representar el coste de moverse entre dos localizaciones concretas.

De esta forma, se evita tener que crear una acción para cada par de localizaciones y define una sola acción Viajar, con los mismos parámetros del ejercicio 1. El efecto de la acción Viajar actualiza el `total-cost` según el valor definido en la función `(= (CosteCamino A B) n)`. Luego, la idea es minimizar el coste del plan con `(:metric minimize (total-cost))`.

Cuando el planificador no garantiza que el plan devuelto sea el más corto, una estrategia para acercarnos a esa optimalidad es usar la búsqueda en profundidad iterativa (IDDFS). La idea del algoritmo es explorar todos los planes posibles de longitud 0, luego de longitud 1, luego 2,... y así sucesivamente, hasta que encuentre uno válido. Esto garantiza que el primer plan que encuentre sea el más corto posible.

Por ejemplo, si quisiéramos formar una comunidad con 1 mago y 2 hobbits y hay 4 ciudades posibles para reunirse. Un plan de 5 pasos podría ser reunirlos en Amon Hen, pero también podría haber un plan más corto de 4 pasos si se encuentran antes. Si dejamos al

planificador sin restricciones, puede que devuelva el de 5. Pero si usamos IDDFS, primero se exploran los planes de 1 acción, luego dos acciones,..., y en cuanto encuentra uno válido de 4 acciones, lo devuelve y se detiene, garantizando que es el más corto.