Intel·ligència Artificial (Laboratorio)

Práctica de Planificación

Documentación

Curso 2022/2023 1Q

Grau en Enginyeria Informàtica - UPC
Departament de Ciències de la Computació

UPC
FIB Facultat d'Informàtica de Barcelona
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Índice general

Capítulo 1

Organización, evaluación y entrega

Esta es la documentación de la práctica de planificación para los alumnos de Inteligencia Artificial del Grado en Informática. En este documento tenéis:

- Los objetivos de aprendizaje de la práctica correspondientes al temario de la asignatura
- La descripción del problema que debéis resolver
- Lo que tenéis que incluir en el informe que deberéis entregar como resultado de la práctica
- La planificación semanal de la práctica incluyendo los objetivos que debéis ir cubriendo cada semana.

La práctica se debe hacer **preferentemente en grupos de 3 personas**. Intentad no hacerla solos ya que os llevará mucho más trabajo.

La práctica se debe desarrollar con **Fast Forward v2.3**, el planificador que se presentará en las clases de laboratorio.

Planificad bien el desarrollo de la práctica y no lo dejéis todo para el último día, ya que no seréis capaces de acabarla y hacer un buen trabajo. En este documento tenéis indicaciones sobre el desarrollo de la práctica que os ayudarán a planificar vuestro trabajo.

La valoración de la práctica dependerá de la calidad del modelado del problema, de la cobertura del problema que hagáis, de las extensiones que abordéis y de la calidad de los juegos de prueba.

La entrega de la documentación será el día **9 de enero** en formato electrónico según las instrucciones que aparecerán en el racó.

Capítulo 2

Objetivos de aprendizaje

El objetivo de esta práctica es enfrentarse a un problema sencillo de síntesis que se puede resolver mediante un planificador en el espacio de estados para que construya la solución.

Los objetivos específicos de esta práctica son:

- Implementar mediante un lenguaje de descripción (PDDL) el dominio (predicados y acciones) y varios ejemplos de problemas (objetos, estados inicial y final)
- Aplicar una metodología de desarrollo basada en prototipaje rápido y diseño incremental.
- Saber escoger juegos de prueba suficientemente representativos para demostrar el funcionamiento del sistema y explicar los resultados.
- Tomar contacto con lenguajes de representación de acciones que se puedan usar con planificadores modernos. Se ha de demostrar cierta comprensión y madurez a la hora de utilizar el lenguaje PDDL.
- Conectar lo que se ha hecho en la práctica de Sistemas Basados en el Conocimiento con lo que puede hacer el planificador.

Capítulo 3

El problema

Estamos en el año 2075 y los primeros colonizadores de marte han establecido una red bases a lo largo de la geografía del planeta que pueden ser de dos tipos, asentamientos (donde viven los colonos) y almacenes (a donde llegan los suministros desde la tierra). El desplazamiento entre estas bases se realiza mediante rovers de transporte que pueden mover suministros y personal especializado. **No es posible siempre ir directamente de una base a otra**, tenemos un mapa (grafo conexo) que nos indica qué movimientos son posibles.

Para cubrir las necesidades de los asentamientos, estos hacen peticiones tanto de personal especializado como de suministros a un organismo centralizado. Cada petición es de una unidad de suministro o una persona, si se quiere más se pueden hacer varias peticiones. Este organismo se encarga de planificar los desplazamientos de los rovers desde la base donde se encuentran aparcados, hasta recoger los suministros y el personal necesarios y finalmente para dejarlos en lugar donde se ha hecho la petición.

Asumiremos para hacer más simples las cosas que **siempre hay más** peticiones de suministros y personal de las que se pueden servir (la vida en marte es muy dura).

Problema básico y extensiones

Asumiremos que en un problema tendremos un conjunto de rovers aparcados en bases (pueden ser más de uno y no tienen porque estar en el mismo sitio), hay un conjunto de peticiones hechas por los asentamientos tanto de personal especializado como de suministros y hay personal disponible en asentamientos y suministros en los almacenes esperando para ser transportados. El número de peticiones de cada cosa es igual o mayor a lo que está disponible.

Nada impide que un rover cargue y descargue varias veces en su recorrido.

En algunos apartados os hará falta usar `metric-ff` ya que necesitaréis atributos numéricos, hacer comparaciones entre ellos que no son solo de igualdad y hacer optimización. Tenéis disponibles en el apartado de laboratorio de la asignatura enlaces a los fuentes de este planificador compilables en diferentes sistemas operativos.

- **Nivel básico**: Se realiza el transporte de todos los suministros y personal para cubrir las peticiones que se puedan cubrir con lo que hay disponible. No hay limitación para la capacidad de carga de los rovers y pueden transportar a la vez tantos suministros y personal como quieran.
- **Extensión 1**: Los rovers pueden transportar a la vez un máximo de dos personas o una carga de suministros (no se pueden mezclar personas y suministros).
- **Extensión 2**: Extensión 1 + Los rovers tienen una capacidad de combustible que se carga al principio del día, de manera que solo pueden hacer un número limitado de movimientos. En cada desplazamiento

entre dos bases el rover gasta una unidad de combustible. Haced una versión en la que no importe cuanto combustible gastan en total los rovers mientras hagan las entregas y otra en la que se minimice el combustible total gastado para poder comparar las soluciones.

- **Extensión 3**: Extensión 2 + las peticiones tienen una prioridad 1, 2 o 3 (siendo 3 la máxima) de manera que buscamos el plan que maximice la prioridad de las peticiones servidas. Haced una versión en la que no importe el total de combustible gastado y otra en la que se optimice una combinación entre prioridades y combustible total en la que se puedan asignar diferentes pesos a cada criterio para poder comparar las soluciones.

Respecto a la optimización de los fluentes, tened en cuenta que al planificador solo le dais los costes de los operadores y no puede hacer una búsqueda exhaustiva (para guiarse usa una estimación heurística independiente del dominio de la longitud de los caminos), por lo que no esperéis que se obtenga siempre la solución óptima.

Según las extensiones que decidáis abordar la nota de la práctica será diferente:

- Nivel básico: la nota máxima es un 6
- Extensión 1: la nota máxima es un 7
- Extensión 2: la nota máxima es un 8.5
- Extensión 3: la nota máxima es un 10

Para resolver los problemas, al generar el grafo de los asentamientos aseguraos de que sea conexo (pero que no sea completamente conectado), tampoco necesita tener muchas conexiones.

Nota extra

Los juegos de prueba los podéis hacer a mano, pero se asignará **un punto extra** a los grupos que hagan un programa (no importa el lenguaje) que pueda generar ficheros con juegos de prueba generados aleatoriamente y los use en las pruebas.

Se asignará **otro punto extra** a los grupos que usen este generador para obtener problemas de tamaño creciente y experimenten como el tiempo de resolución evoluciona con el tamaño del problema. Asumiendo un tamaño de grafo fijo podéis estudiar qué pasa si aumenta el número de alguno de los otros elementos del problema (no hace falta que los probéis todos):

- el número de rovers
- el número de peticiones
- el número de personal/suministros a transportar

Tened en cuenta que el `metric-ff` tiene una limitación en el número de líneas que puede tener un fichero de problema (alrededor de 200), pero que no os hace falta probar tamaños muy grandes de problema para ver la tendencia.

Documentación a entregar

La documentación debe incluir:

- Un documento en el que se describa, de forma razonada
  - La forma en la que se ha modelado el dominio (variables, predicados y acciones)
  - La forma en la que se modelan los problemas a resolver (objetos, estado inicial y final)
  - Una breve explicación de cómo habéis desarrollado los modelos (de una sola vez, por iteraciones)
  - Un conjunto de problemas de prueba no triviales (mínimo 2), explicando su resultado. Explicad especialmente qué diferencia a las soluciones que se obtiene en cada juego de pruebas con cada extensión.
- Código en pddl del dominio que habéis modelado para cada extensión y los problemas de prueba.
- Un fichero que recolecte la traza de la resolución de los problemas de prueba.

Capítulo 4

Guión de la práctica

Primera semana: Fast Forward/Enunciado/creación del primer prototipo (19 de diciembre)

Esta primera semana la deberéis dedicar a leer el enunciado, a hacer un modelo inicial de dominio y problema y a crear un modelo en PDDL que llegue al nivel básico.

Esta semana se os explicará el funcionamiento del planificador Fast Forward. Es importante que leáis la documentación sobre PDDL y Fast Forward que se os dará, miréis los ejemplos que tenéis e intentéis ejecutarlos.

Tened en cuenta que modelar dominios en PDDL necesita una forma de pensar algo diferente a la que estáis acostumbrados con los lenguajes imperativos y lógicos, por lo que es importante que empecéis cuanto antes a ver cómo funciona.

Si tenéis planeada alguna de las extensiones deberíais de poneros ya con ellas a media semana, ya que la última semana deberéis dedicar algo de tiempo a la documentación y a las pruebas.

En esta práctica es importante planificar vuestro trabajo, no lo dejéis todo para el último momento.

Segunda semana: Prototipo definitivo / Juegos de prueba y documentación (26 de diciembre/2 de enero)

En esta semana deberíais tener ya un planificador que, como mínimo, es capaz de crear planes en el nivel básico. A principios de la semana ya deberíais haber fijado todas la extensiones que queréis intentar hacer y tenerlas algo avanzadas a media semana.

Mirad los ejemplos de problemas modelados en PDDL que tenéis en la web de la asignatura y en otras páginas en Internet para inspiraros.

Deberéis plantearos los casos que queréis probar y mirar que los resultados que esperáis sean los correctos. Haced una lista de casos pensando los diferentes escenarios que es capaz de resolver vuestro sistema.

Pensad que los casos han de ser suficientemente variados tanto en lo que respecta a elementos que intervienen como su complejidad. Tened en cuenta que estos casos os servirán de juegos de prueba, por lo que estáis matando dos pájaros de un tiro. Aprovechad para guardar los resultados y documentarlos.

También deberíais ser capaces de explicar los resultados que obtenéis en función del conocimiento que habéis programado.

Las pruebas deberíais documentarlas adecuadamente explicando cual es el escenario de la prueba y

cuales son los resultados que da el sistema.

El resto de la documentación debería explicar todo el proceso de desarrollo y los diferentes prototipos que habéis creado por el camino.

No hace falta que esperéis al último día para entregar. Si acabáis la práctica y la documentación antes podéis entregarla ya durante la semana.

Capítulo 5

Rúbrica de evaluación

Esta es la rúbrica de evaluación de la práctica. La corrección se hará según estos criterios y siguiendo las pautas que se detallan para cada nivel de evaluación.

Deberéis seguir estos criterios a la hora de escribir vuestra documentación y explicar qué habéis hecho en el desarrollo de la práctica y como lo habéis hecho.

Valoración / Criterio	Mal	Regular	Bien
Dominio	▪ El dominio se representa de manera incompleta o inadecuada (predicados innecesarios)	▪ Se representa completa y adecuadamente las características del dominio ▪ La explicación de la representación del dominio es superficial	▪ Se representa completa y adecuadamente las características del dominio ▪ Se explica detalladamente el significado de cada predicado y se justifica su necesidad
Operadores	▪ El conjunto de operadores es inadecuado o incompleto	▪ El conjunto de operadores es adecuado y completo ▪ La explicación/justificación de los operadores es superficial	▪ El conjunto de operadores es adecuado y completo ▪ Se explica cada operador y se justifica detalladamente su necesidad para la resolución del problema
Juegos de prueba	▪ Juegos de prueba inadecuados para el problema planteado	▪ Juegos de prueba adecuados ▪ No se justifica la elección de los juegos de prueba	▪ Juegos de prueba adecuados ▪ Se justifica la elección de los juegos de prueba ▪ Se explica la solución obtenida
Completado de los niveles	▪ La solución propuesta para los diferentes niveles es inadecuada o incompleta		▪ La solución propuesta para los diferentes niveles es adecuada y completa