Documentación de la práctica de Planificación

Laboratorio de Inteligencia Artificial

2º Cuatrimestre - curso 2017/2018

Grau en Informàtica

Departament de Ciències de la Computació

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 **11 de junio** 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

La agencia de viajes *al fin del mundo y mas allá* quiere una versión más sencilla de su sistema de recomendación de viajes para su página de web y ha decidido resolverlo mediante un planificador. La idea es obtener a partir de ciertas restricciones del usuario y su base de datos de vuelos y de hoteles un plan de viaje.

Problema básico y extensiones

En todos los apartados os hará falta usar `metric-ff` ya que como mínimo necesitaréis atributos numéricos y hacer comparaciones entre ellos que no son solo de igualdad. Os lo podéis bajar de http://fai.cs.uni-saarland.de/hoffmann/metric-ff.html y lo deberéis compilar para el SO que utilicéis.

Para todos los problemas asumiremos que el planificador elige una ciudad de inicio (no hay que fijarla de antemano) y conecta varias ciudades mediante un camino (definido por vuelos) sin repetir ninguna. Para simplificar, asumiremos que los vuelos entre la ciudad origen del cliente y las ciudades inicio y final del recorrido no forman parte del problema.

- **Nivel básico**: Dado el conjunto de ciudades disponibles, los hoteles en cada ciudad y los vuelos que conectan las ciudades, queremos que, dado un mínimo número de ciudades a visitar, nos indique un hotel en el que alojarse y los vuelos que hay que tomar para viajar entre las ciudades.
- **Extensión 1**: Nivel básico + Se indica cuál es el mínimo y máximo número de días que se ha de estar en las ciudades y el mínimo número de días que ha de tener el recorrido y obtenemos un plan que respeta esas restricciones y nos dice cuantos días se está en cada ciudad.
- **Extensión 2**: Extensión 1 + Cada ciudad tiene un interés representado como un valor entero (por ejemplo de 1 a 3, siendo 1 el máximo interés). Queremos un plan que maximice (tal como es el interés en realidad lo que queremos es minimizar) el interés de las ciudades que visitamos.
- **Extensión 3**: Extensión 1 + Los vuelos y los hoteles tiene un precio marcado y queremos minimizar el precio total del plan, pero indicando un precio mínimo y un precio máximo total. El rango de precios puede ser cualquiera, pero sed coherentes al imponer el precio mínimo/máximo para que haya solución.
- **Extensión 4**: Queremos unir las extensiones 2 y 3 de manera que el plan optimice el interés de las ciudades visitadas y el precio total del recorrido. Dado que el interés y el precio tienen unidades muy diferentes, deberéis usar ponderaciones en la métrica que optimicéis, comprobando el efecto de usar diferentes valores (básicamente ver que salen planes diferentes dando ponderaciones diferentes).

Para el nivel básico y las extensiones 1 y 2, como no hay nada que distinga vuelos y hoteles, con tener un vuelo que conecte cada par de ciudades y un hotel por ciudad es suficiente. Para las otras extensiones es necesario tener vuelos y hoteles con diferentes precios.

Para simplificar el hacer los juegos de prueba podemos suponer que hay al menos un vuelo de ida y vuelta entre cada par de ciudades, pero eso puede multiplicar el espacio de búsqueda si hay muchas ciudades. Otra opción es que no todas las conexiones sean posibles (es lo que pasa en realidad), pero hay que asegurarse que el grafo sea mínimamente conexo para que exista alguna solución.

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 5
- Extensión 1: la nota máxima es un 6
- Extensión 2 o 3: la nota máxima es un 7
- Extensión 2 y 3: la nota máxima es un 8.5
- Extensión 4: la nota máxima es un 10

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 lo usen en las pruebas.

Se asignará otro punto extra a los grupos que usen este generador para obtener problemas de tamaño creciente y experimenten como evoluciona el tiempo de resolución en la extensión 3 a medida que se aumenta el numero de ciudades. Para hacer este estudio no podréis tener un par de vuelos por cada ciudad a partir de cierto tamaño de problema, así que tendréis que generar un grafo de vuelos con solo un porcentaje de conexiones del total posible. Tened en cuenta también 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 por cada extensión (mínimo 2), explicando para cada uno que es lo que intentan probar y su resultado. En las extensiones en las que hay que ponderar diferentes criterios comentad el efecto que tienen sobre las soluciones. Podéis ir partiendo de los juegos de prueba para el nivel básico e ir añadiendo los elementos que cada extensión requiera.

- 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 (28 de mayo)

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 (4 de junio)

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 las 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 11 de junio 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.

Valoraci / 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