2022/2023 2Q

Práctica de Planificación

Intel·ligència Artificial

Departament de Ciències de la Computació

Grau en Enginyeria Informàtica - UPC

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.
- Rúbrica de evaluación de la práctica

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 **12 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.

Respecto a la **evaluación**, tenéis disponible una rúbrica que indica los criterios que se usarán para valorar la práctica y una descripción de cada uno de los niveles de valoración para cada criterio.

Capítulo 3

El problema

Estamos a cargo de la distribución de tareas de un proyecto informático de gran envergadura y queremos hacerlo de manera automatizada usando planificación automática. El proyecto se compone de T tareas de programación que tienen asignada una dificultad de menor a mayor (1, 2, 3) y un tiempo estimado estimado de realización (en horas).

Tenemos a nuestra disposición P programadores, cada uno de ellos tiene asociada una habilidad de menor a mayor (1, 2, 3) que nos indica la dificultad que puede resolver. Asumiremos que a un programador se le puede asignar a una tarea de dificultad como mucho una unidad por encima de su habilidad. Si la dificultad de la tarea es superior a su habilidad, el tiempo para realizarla se incrementa en dos horas.

Los programadores tienen también asociada una calidad (1, 2), de manera que si un programador de calidad 1 realiza una tarea, hará falta una tarea adicional de revisión de 1 hora y si es de calidad 2 hará falta una tarea de revisión de 2 horas. Esta tarea será de la misma dificultad que la que se realizó y no puede asignarse a la misma persona. Estas tareas, cuando se resuelven, no generan ninguna tarea adicional y se asignan con el mismo criterio, pero el tiempo que se tarda en resolverlas no se incrementa por la habilidad del programador asignado.

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. Tenéis disponibles en el apartado de laboratorio de la asignatura enlaces a los fuentes de este planificador compilables en diferentes sistemas operativos.

- **Nivel básico**: Dado el conjunto de tareas y programadores disponibles, obtener una asignación de todas las tareas a programadores sin tener en cuenta la tarea adicional que genera cada asignación a un programador.
- **Extensión 1**: Nivel básico + ahora se tiene en cuenta la tarea adicional que genera la calidad de los programadores.
- **Extensión 2**: Extensión 1 + ahora nos interesa minimizar el tiempo total que se usa en resolver todas las tareas (suma de las horas).
- **Extensión 3**: Extensión 2 + ahora también nos interesa poder tener el máximo de personas trabajando para que se puedan hacer más cosas en paralelo, de manera que queremos limitar el número de tareas que podemos asignar a una persona a 2.
- **Extensión 4**: Extensión 3 + aunque queremos tener más personas para que se hagan cosas en paralelo, también queremos limitarlo. Además de mantener el límite en el número de tareas, queremos hacer una versión en la que se optimice la suma ponderada entre el número de personas que están haciendo tareas y el tiempo total que se tardan en resolverlas. ¿Cual es la ponderación más adecuada?

Tened cuidado al elegir los juegos de prueba de manera que haya suficientes programadores para que se puedan resolver las tareas.

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 de la longitud de los caminos que es independiente del dominio), 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: la nota máxima es un 7
- Extensión 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.

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 2 a medida que se aumenta el numero de tareas y programadores. El fast forward tiene un límite para las líneas que puede tener un fichero de problema (alrededor de 200 el parser deja de funcionar correctamente) así que no se podrán probar tamaños muy grandes, pero os deberíais poder hacer una idea del crecimiento del tiempo.

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 qué es lo que intentan probar y su resultado. Podéis partir 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 (29 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 (25 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 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