

Documentación de la práctica de Planificación

Laboratorio de Inteligencia Artificial

1^{er} Cuatrimestre - curso 2017/2018

Grau en Informàtica

Departament de Ciències de la Computació



FIB

Facultat d'Informàtica
de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Índice general

1. Organización, evaluación y entrega	2
2. Objetivos de aprendizaje	3
3. El problema	4
4. Guión de la práctica	6
5. Rúbrica de evaluación	8

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 **15 de enero** en formato electrónico según las instrucciones que aparecerán en el racó.

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.

El problema

Una central de reservas de un hotel necesita un sistema capaz de asignar las peticiones de reserva que se reciben a habitaciones, siguiendo diferentes criterios y restricciones. Para simplificar consideraremos que las reservas son únicamente para un mes concreto (30 días).

Una habitación esta descrita por su identificador y el número de personas que puede alojar (entre 1 y 4). Una reserva esta descrita por su identificador, el número de personas en la reserva (entre 1 y 4) y los días de inicio y final de la reserva (entre el 1 y el 30).

Problema básico y extensiones

En todos los apartados os hará falta usar `metric-ff` ya que como mínimo necesitareis atributos numéricos y hacer comparaciones entre ellos que no son solo de igualdad.

- **Nivel básico:** Se obtiene una asignación de reservas a habitaciones en las que caben las personas de la reserva (la habitación puede ser más grande) y no hay solapamiento en las ocupaciones. Si no se pueden asignar todas las reservas no se asigna ninguna.
- **Extensión 1:** Se obtiene una asignación de reservas a habitaciones en las que caben las personas de la reserva (la habitación puede ser más grande) y no hay solapamiento en las ocupaciones. Se optimiza el número de reservas asignadas, pueden quedar reservas sin asignar.
- **Extensión 2:** Extensión 1 + En las peticiones se puede indicar la orientación preferente de la habitación (N/S/E/O) y se quiere obtener una asignación que optimice las peticiones que están en una habitación con la orientación solicitada. En este caso deberéis optimizar la combinación del criterio de la extensión 1 con el de la orientación, teniendo en cuenta que es peor no asignar una reserva que el asignarle una orientación incorrecta.
- **Extensión 3:** Extensión 1 + Se obtiene una asignación de reservas a habitaciones, pero además se minimiza el desperdicio de plazas. Es decir, consideramos mejor asignar las reservas a la habitación con tamaño más cercano al número de personas de la reserva. En esta extensión también tendréis que combinar el criterio de la extensión 1 con el desperdicio de plazas, considerando el diferente peso que deben tener.
- **Extensión 4:** Extensión 3 + Se obtiene una asignación en la que también se optimiza el número de habitaciones que se ocupan durante el mes. Es decir, son mejores soluciones en las que las reservas se asignan a un menor número de habitaciones diferentes. Aquí tendréis que considerar los pesos de los tres criterios, considerando que es peor no asignar reservas y tiene prioridad no abrir habitaciones respecto a desperdiciar plazas.

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.

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. 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 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.
- 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.

Guión de la práctica

Primera semana: Fast Forward/Enunciado/creación del primer prototipo (18 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 ponerlos 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 (8 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 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 15 de enero para entregar. Si acabáis la práctica y la documentación antes podéis entregarla ya durante la semana.

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.

Criterio	Valoración		Bien
	Mal	Regular	
Dominio	<ul style="list-style-type: none"> El dominio se representa de manera incompleta o inadecuada (predicados innecesarios) 	<ul style="list-style-type: none"> Se representa completa y adecuadamente las características del dominio La explicación de la representación del dominio es superficial 	<ul style="list-style-type: none"> Se representa completa y adecuadamente las características del dominio Se explica detalladamente el significado de cada predicado y se justifica su necesidad
Operadores	<ul style="list-style-type: none"> El conjunto de operadores es inadecuado o incompleto 	<ul style="list-style-type: none"> El conjunto de operadores es adecuado y completo La explicación/justificación de los operadores es superficial 	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Juegos de prueba inadecuados para el problema planteado 	<ul style="list-style-type: none"> Juegos de prueba adecuados No se justifica la elección de los juegos de prueba 	<ul style="list-style-type: none"> Juegos de prueba adecuados Se justifica la elección de los juegos de prueba Se explica la solución obtenida
Completado de los niveles	<ul style="list-style-type: none"> La solución propuesta para los diferentes niveles es inadecuada o incompleta 		<ul style="list-style-type: none"> La solución propuesta para los diferentes niveles es adecuada y completa