

Práctica de Planificación

Intel·ligència Artificial



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



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	7
5. Rúbrica de evaluación	9

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 ayudará 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 **10 de junio de 2024** 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

Nuestro cliente, después de quedar satisfecho con el sistema de recomendación de ejercicios que le hemos construido en CLIPS, nos pide ahora una herramienta sencilla que, basándose en los ejercicios que el usuario esta realizando en este momento y su nivel de dificultad (del 1 al 10), les haga un plan de entrenamiento físico quincenal con el objetivo de conseguir subir de nivel en un conjunto de ejercicios escogidos. El plan de entrenamiento ha de tener en cuenta que no se puede realizar un ejercicio con un nivel X de dificultad si no se ha realizado el mismo ejercicio con nivel de dificultad $X-1$ en un día anterior, y que hay ejercicios que se deben de hacer antes de otros. Definiremos las siguientes relaciones entre ejercicios:

- Ejercicios precursores: un ejercicio es precursor de otro si se tiene que hacer justo antes del otro ejercicio en el mismo día de entrenamiento, sin tener un ejercicio intercalado entre ellos. Es una restricción estricta que no se puede romper.
- Ejercicios preparadores: son ejercicios que se han de hacer antes de un cierto ejercicio en el mismo día de entrenamiento pero no tienen porque ir inmediatamente antes (un ejemplo son los estiramientos, que preparan para varios ejercicios de la sesión de entrenamiento). Es una restricción estricta que no se puede romper.
- Ejercicios nivel +1: sirve para modelar la evolución en el nivel de dificultad de un ejercicio según pasan los días. Para cada ejercicio podremos subir un nivel $N + 1$ solo si ese ejercicio se ha realizado en algun dia anterior en el nivel N .

Por lo tanto al sistema se le ha de dar conocimiento sobre:

- los ejercicios que se pueden hacer en el gimnasio
- los ejercicios predecesores a un ejercicio
- los ejercicios preparadores de un ejercicio
- los ejercicios que el usuario ya esta haciendo, y en que nivel los hace
- los niveles incrementados de algunos de los ejercicios a los que se quiere llegar al final de la quincena.

El resultado es un plan de entrenamiento que

- refleja los ejercicios mínimos que el usuario ha de hacer para llegar a los niveles objetivo (esto es importante: el planificador no tiene que rellenar el plan con ejercicios extra, solo los imprescindibles para llegar al dia 15 al nivel deseado. Si sobran huecos se entiende que se le da al usuario libertad para hacer otros ejercicios suplementarios al plan de entrenamiento).

- para cada ejercicio en el plan, indica en que día ha de hacerlo y en que nivel para llegar a los objetivos quincenales
- para todos los ejercicios del plan se cumple en todo momento que sus predecesores se hacen justo antes en el mismo día, sin ningún ejercicio intercalado por en medio
- para todos los ejercicios del plan se cumple en todo momento que sus ejercicios preparadores se hacen antes en el mismo día (pero puede haber ejercicios intercalados)
- para todo ejercicio del plan con un nivel N se cumple en todo momento que el mismo ejercicio ha sido realizado en algún día anterior con nivel N o $N-1$.

Problema básico y extensiones

- **Nivel básico:** En el plan de entrenamiento todos los ejercicios tienen 0 o 1 preparadores y ningún predecesor. El planner es capaz de encontrar un plan para poder llegar a realizar los ejercicios en el nivel objetivo al final de la quincena encadenando ejercicios y subiendo el nivel si hace falta según van pasando los días, teniendo en cuenta que cada ejercicio tiene solo uno o ningún ejercicio preparador (los estiramientos no tienen ningún preparador) y que para todo ejercicio del plan con un nivel N se cumple en todo momento que el mismo ejercicio ha sido realizado en algún día anterior con nivel N o $N-1$.
- **Extensión 1:** Los ejercicios pueden tener de 0 a N preparadores pero ningún predecesor. El planner es capaz de construir un plan para poder llegar a realizar los ejercicios en el nivel objetivo encadenando ejercicios y subiendo el nivel si hace falta según van pasando los días, teniendo en cuenta que para todo ejercicio que pertenece al plan, todos sus ejercicios preparadores pertenecen al plan y están antes en el mismo día.
- **Extensión 2:** Extensión 1 + los ejercicios pueden tener 0 o 1 ejercicio predecesor. El planner es capaz de construir un plan para poder llegar a realizar los ejercicios en el nivel objetivo encadenando ejercicios y subiendo el nivel si hace falta según van pasando los días donde para todo ejercicio que pertenece al plan, todos sus ejercicios precursores y preparadores están colocados el mismo día y para todo ejercicio del plan con un nivel N se cumple en todo momento que el mismo ejercicio ha sido realizado en algún día anterior con nivel N o $N-1$.
- **Extensión 3:** El planificador controla que no se coloquen más de 6 ejercicios por día.
- **Extensión 4:** Los ejercicios tienen asignados el tiempo (en minutos) que se tarda en realizarlos (incluyendo el tiempo de descanso necesario después del ejercicio). El planificador controla que en el plan generado no se superen los 90 minutos al día.

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

- Nivel básico: la nota máxima es un 6
- Nivel básico + extensión 1: la nota máxima es un 7
- Nivel básico + extensión 2: la nota máxima es un 8
- extensión 3: +1 punto en la nota
- extensión 4: +2 puntos en la nota

En principio las extensiones 3 y 4 son mutuamente excluyentes, pero si un grupo presenta una versión del prototipo que cubre la extensión 3 y otra versión que cubre la extensión 4, puede obtener +3 puntos (con lo que la nota de la práctica puede llegar a 11).

Para la modelar el problema hasta la extensión 3 probablemente lo podréis hacer sin utilizar fluentes y podréis calcular el plan con FF, pero para la extensión 4 depende de como modeléis el problema, sí os harán falta usar fluentes y **Metric-FF** para calcular el plan.

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.

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 como habéis desarrollado los modelos (de una sola vez, por iteraciones)
 - El conjunto de problemas de prueba (mínimo 2), explicando para cada uno que 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.

Guión de la práctica

Primera semana: Fast Forward/Enunciado/creación del primer prototipo (27 de mayo a 2 de junio)

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 deberías 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 (3 de junio a 9 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 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 hasta el último día 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