

# Documentación de la práctica de Planificación

Laboratorio de Inteligencia Artificial

1<sup>er</sup> Cuatrimestre - curso 2019/2020

Grau en Informàtica

Departament de Llenguatges i Sistemes Informàtics



**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.
- 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 **9 de enero de 2020** 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.

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.

## El problema

Nuestro cliente, después de quedar satisfecho con el sistema de recomendación de libros que le hemos construido en CLIPS, nos pide ahora una herramienta sencilla que, basándose en los libros que ya han leído y los libros que quieren leer durante el próximo año, les haga un plan de lectura mensual, en el que se le recomiende en qué orden han de leer los libros, intentando balancear un poco el número de páginas que le tocaría leer cada mes. El plan de lectura ha de tener en cuenta que a veces hay dependencias entre libros:

- Libros predecesores: es típico de sagas o de historias divididas en varios libros. Llamaremos libros predecesores a aquellos libros que se han de leer antes de un libro para poder disfrutarlo mejor. Por ejemplo, si el usuario quiere leer la 4ª entrega de Harry Potter y solo ha leído el primer libro de la saga, el planificador ha de incluir en el plan de lectura el segundo y tercer libro, que preceden al cuarto.
- Libros paralelos: esto ocurre en aquellos casos en que los libros pertenecen a universos ficticios donde las historias no solo pasan una después de la otra, sino también en paralelo, por lo que es bueno leer las historias más o menos en el orden en el que ocurren. Este fenómeno es muy típico del cómic americano desde los años 60 (crossovers), pero también pasa en los libros de universos como Star Wars o Star Trek.

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

- los libros del catálogo
- los libros predecesores a un libro
- los libros paralelos a un libro
- los libros que el usuario ya ha leído
- los libros que el usuario quiere leer

El resultado es un plan de lectura que

- refleja los libros mínimos que el usuario ha de leer
- para cada libro, indica en que mes ha de leerlo
- para todos los libros del plan se cumple en todo momento que sus predecesores se leen en meses anteriores
- para todos los libros del plan se cumple en todo momento que sus paralelos se leen en el mismo mes o en el mes anterior (o en el mes siguiente, para la relación simétrica).

## Problema básico y extensiones

- **Nivel básico:** En el plan de lectura todos los libros tienen 0 o 1 predecesores y ningún paralelo. El planner es capaz de encontrar un plan para poder llegar a leer los libros objetivo encadenando libros, donde cada libro tiene solo uno o ningún predecesor.
- **Extensión 1:** Los libros pueden tener de 0 a N predecesores pero ningún paralelo. El planner es capaz de construir un plan para poder llegar a leer los libros objetivo, donde para todo libro que pertenece al plan, todos sus libros predecesores pertenecen al plan y están en meses anteriores.
- **Extensión 2:** Extensión 1 + los libros pueden tener de 0 a M libros paralelos. El planner es capaz de construir un plan para poder llegar a leer los libros objetivo, donde para todo libro que pertenece al plan, todos sus libros paralelos pertenecen al plan y están en el mismo mes o en meses anteriores.
- **Extensión 3:** Los libros tienen además un número de páginas. El planificador controla que en el plan generado no se superen las 800 páginas al mes.

Según como modeléis el problema, si usáis atributos numéricos y hacéis comparaciones entre ellos que no sean solo de igualdad, os hará falta utilizar `metric-ff`. Os lo podéis bajar de <http://fai.cs.uni-saarland.de/hoffmann/metric-ff.html> (`Metric-FF.tgz`) y lo deberéis compilar para el SO que utilizéis.

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: 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 los use como juegos de prueba. Pensad que los problemas son conjuntos de grafos dirigidos acíclicos (DAG, en inglés). También podéis aprovechar que un árbol es un caso particular de DAG.

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 libros y sus dependencias.

Atención: 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.

## Guión de la práctica

### **Primera semana: Fast Forward/Enunciado/creación del primer prototipo (16 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 (30 de diciembre)**

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