

# **PRÁCTICA 3- PDDL**

**Yoel Cabo  
Patricia Sampedro  
Octavi Pascual**

## 1. Modelaje del dominio

### 1.1. Variables

### 1.2. Predicados

### 1.3. Acciones

### 1.4. Representación del problema

## 2. Extensiones

### 2.1. Nivel Básico

### 2.2. Extensión 1

### 2.3. Extensión 2

#### 2.3.1. Generador

##### 2.3.1.1. ¿Cómo se ha hecho?

##### 2.3.1.2. ¿Cómo ejecutarlo?

### 2.4. Extensión 3

## 3. Juegos de prueba

### 3.1. Nivel básico

#### 3.1.2. Prueba Problem1

##### 3.1.1.1. Resultado esperado

##### 3.1.1.2. Entrada simplificada

##### 3.1.1.4. Salida

##### 3.1.1.5. Resultado

#### 3.1.2. Prueba Problem2

##### 3.1.2.1. Resultado esperado

##### 3.1.2.2. Entrada simplificada

##### 3.1.2.3. Salida

##### 3.1.2.4. Resultado

##### 3.1.2.5. Conclusión

### 3.2. Extensión 1

#### 3.2.1. Prueba Problem1

##### 3.2.1.1. Resultado esperado

##### 3.2.1.2. Entrada simplificada

##### 3.2.1.1.3. Salida

##### 3.2.1.4. Resultado

#### 3.2.2. Prueba Problem2

##### 3.2.2.1. Resultado esperado

##### 3.2.2.2. Entrada simplificada

##### 3.2.2.3. Salida

##### 3.2.2.4. Resultado

#### 3.2.3. Prueba Problem2bis

##### 3.1.3.1. Salida

##### 3.2.3.2. Resultado

#### 3.2.5. Prueba Problem3

[3.2.5.1. Resultado esperado](#)

[3.2.5.2. Entrada simplificada](#)

[3.2.5.3. Salida](#)

[3.2.5.4. Resultado](#)

[3.3.Extensión 2](#)

[3.3.1. Prueba Ext2-1](#)

[3.3.1.1. Resultado esperado](#)

[3.3.1.2. Entrada simplificada](#)

[3.3.1.3. Salida](#)

[3.3.1.4. Resultado](#)

[3.3.2. Prueba Ext2-2](#)

[3.3.2.3. Salida](#)

[3.3.2.4. Resultado](#)

[3.3.Extensión 3](#)

[3.3.1. Problem1](#)

[3.3.2. Problem2](#)

# 1. Modelaje del dominio

## 1.1. Variables

Para este problema las variables que hemos usado son las siguientes: libro y mes.

Libro representa un libro, el concepto central del dominio que queremos representar.

Mes representa un mes real o fantasma. Los meses reales son los doce meses del año (de enero a diciembre). Existen dos meses fantasma, que se sitúan después de diciembre. De alguna manera podríamos decir que se corresponden a enero y febrero del año siguiente. Más adelante explicaremos detalladamente por qué hemos tomado la decisión de añadir dos meses fantasma.

## 1.2. Predicados

Los predicados que hemos utilizado son:

- (leído ?l - libro)

El libro l ha sido leído por el lector. Remarcar que esto significa que se ha leído hace dos meses o más.

- (leyendo ?l - libro)  
El libro l está siendo leído este mes.
- (leyendo\_mes\_anterior ?l - libro)  
El libro l ha sido leído el mes anterior.
- (deseado ?l - libro)  
El lector ha indicado que desea leer el libro l.
- (predecesor ?l - libro ?lp - libro)  
El predecesor del libro l es el libro lp. Si queremos leer l antes tendremos que haber leído lp.
- (paralelo ?l1 - libro ?l2 - libro)  
Indicaa si l1 y l2 son libros paralelos
- (mes\_actual ?m - mes)  
El mes en el que estamos es m.
- (mes\_siguiete ?m1 - mes ?m2 - mes)  
El mes siguiente de m1 es m2. Por ejemplo, el mes siguiente de enero es febrero.
- (es\_fantasma ?m - mes)  
Indica si un mes es fantasma. Como ya hemos introducido, tendremos 2 meses fantasma, los otros (de enero a diciembre) no lo son.

A primera vista puede parecer innecesario representar con tres predicados distintos el concepto de leer un libro. Separamos el hecho de haber leído el libro, haber leído el libro en el mes anterior y estar leyendo el libro en el mes actual. Esto es así para poder soportar los libros paralelos: en todo momento debemos saber qué libros fueron leídos el mes anterior para poder asignar al mes actual todos los otros libros paralelos a ellos.

El predicado de libro deseado es ineludible ya que debemos de representar de alguna forma qué libros quiere leer el lector durante el año. De la misma forma, los predicados predecesor y paralelo nos permiten representar las dependencias que existen entre libros. Finalmente destacar que, aunque el concepto de libro paralelo sea simétrico, solo utilizaremos un predicado para expresar esta relación entre 2 libros (es decir, no duplicaremos el predicado).

En último lugar están los predicados de meses. Hay que indicar en qué mes estamos para saber cuando leemos un libro. También hay que guardar de alguna forma la relación entre los meses. En nuestro caso hemos optado por el predicado `mes_siguiente`.

El predicado `mes_fantasma` es necesario para poder simplificar nuestras acciones. Teníamos dos opciones: o realizar una acción especial para finalizar nuestro plan o inventar dos meses fantasma. Esta situación se debe a que comprobamos que nuestro plan es válido cuando intentamos pasar de mes. Por lo tanto no podemos acabar con algún libro que se esté leyendo en el mes actual. Se implementaron ambas opciones y de forma empírica se observó que el planner encontraba una solución más rápidamente con la segunda opción. Acabar era una acción muy similar a pasar de mes, con la única diferencia que una vez se realizaba dicha acción ya no se podían leer libros. El problema era que la acción tenía muchos existenciales y era muy costoso para el planner ir probando si en un punto se había acabado o no. Por estas razones se optó por implementar dos meses fantasma incluso si no es del todo elegante.

## 1.3. Acciones

Las acciones que hemos utilizado son:

- Acción: leer
- Parámetros: ?l - libro ?m - mes
- Precondiciones:
  - $\neg(\text{leído } l)$
  - $\neg(\text{leyendo\_mes\_anterior } l)$
  - $\neg(\text{leyendo } l)$

- $\neg(\text{es\_fantasma } m)$
- $(\text{mes\_actual } m)$
- $\neg \exists \text{ libro } p \text{ tal que:}$ 
  - $(\text{predecesor } l \ p) \wedge \neg(\text{leído } p) \wedge \neg(\text{leyendo\_mes\_anterior } p)$
- Efectos:
  - $(\text{leyendo } l)$

Esta acción permite asignar un libro en un mes al lector. Nos aseguramos que el libro no haya sido leído y que estamos en un mes válido. También nos aseguramos que todos los predecesores del libro hayan sido leídos.

El efecto es que se pasa a leer el libro.

Es importante remarcar que no hemos incluido el predicado deseado en la precondition. Esto podría parecer inadecuado ya que solo tenemos que leer los libros que desea el lector. No obstante, si el lector desea leer el último libro de una saga y no desea los anteriores, tenemos que hacer que desee estos libros de la saga. Se probó realizar una nueva acción llamada desear, que deseaba un libro si su sucesor era deseado. El problema de esta acción es que provocaba que se intentara desear cada libro y el programa se volvía muy ineficiente, más que realizarlo de la forma que aquí mostramos.

Por lo tanto, como también debemos leer los predecesores de un libro deseado, no podemos imponer que un libro sea deseado para leerlo.

- Acción: pasar\_de\_mes
- Parámetros: ?m - mes ?m2 - mes
- Precondiciones:
  - $(\text{mes\_actual } m)$
  - $(\text{mes\_siguiente } m \ m2)$
  - $\neg \exists \text{ libro } l \text{ tal que:}$ 
    - $(\text{leyendo\_mes\_anterior } l) \wedge \neg \exists \text{ libro } p \text{ tal que:}$ 
      - $(\text{paralelo } l \ p) \wedge \neg(\text{leyendo } p) \wedge \neg(\text{leyendo\_mes\_anterior } p) \wedge \neg(\text{leído } p)$
- Efectos:
  - $\neg(\text{mes\_actual } m)$
  - $(\text{mes\_actual } m2)$
  - $\forall \text{ libro } l \text{ tal que } \text{leyendo\_mes\_anterior}(l):$

- $(\text{leído } l) \wedge \neg(\text{leyendo\_mes\_anterior}(l))$
- $\forall \text{libro } l \text{ tal que } \text{leyendo}(l):$ 
  - $(\text{leyendo\_mes\_anterior } l) \wedge \neg(\text{leyendo } l)$

Esta acción nos permite pasar de mes. Es una operación necesaria ya que sino siempre nos quedaríamos en el mismo mes, en enero. La precondition de esta acción sirve para asegurarnos de que la restricción de libros paralelos se cumple.

Los efectos son que los libros que se estaban leyendo pasan a ser leyendo\_mes\_anterior y los libros que se estaban leyendo el mes anterior pasan a ser leídos.

## 1.4. Representación del problema

Para representar el problema primero definimos todos los objetos. En nuestro caso los libros y los meses. Recordar que hay 12 meses normales (de enero a diciembre) y 2 meses fantasma (f1 y f2).

En el apartado de inicializaciones definimos los predicados.

Los que tiene relación con los meses:

- el mes actual es enero
- el mes siguiente de cada mes
- los meses fantasma

Los que tienen relación con los predecesores:

- el libro  $l$  tiene como predecesor  $p$  ( $p$  se tiene que leer antes que  $l$ )

Los libros que ya hemos leído:

- hemos leído el libro  $l$

Los libros que el lector desea:

- el lector desea el libro  $l$



Finalmente el apartado de goal queda muy sencillo. Queremos que todos los libros que desea el lector sean leídos.

Es decir:

$$\forall \text{ libro } l \text{ tal que } l \text{ deseado} \Rightarrow l \text{ leído}$$

En este punto se podría pensar que hubiera sido mucho más sencillo modificar este objetivo y admitir también libros en estado leyendo o leyendo\_mes\_anterior. De esta manera no haría falta usar meses fantasma. El problema es que nosotros realizamos la comprobación de que nuestro plan es correcto al pasar de mes y por lo tanto esta solución no sería correcta.

En resumen, concluimos que la utilización de solamente dos acciones ha condicionado la elección de los predicados. De todas formas, creemos que reducir al máximo el número de acciones afectará positivamente en el tiempo de ejecución del programa. Como hemos explicado, las pruebas realizadas apuntan a la confirmación de este hecho.

## 2. Extensiones

### 2.1. Nivel Básico

En el nivel básico se nos pide realizar un plan de lectura con las siguientes características:

- todos los libros tienen ningún o un único predecesor
- no hay ningún libro paralelo

El objetivo consiste en leer los libros en un orden en el que no se lea un libro antes que su predecesor.

Recordemos que para todo libro su predecesor debe leerse en meses anteriores a él.

Este problema inicial es sencillo pero debe plantearse muy cuidadosamente ya que será nuestro prototipo inicial que más adelante extenderemos. Por esta razón vamos a intentar estructurarlo de manera que sea fácilmente escalable.

Realizando un análisis de este problema, nos damos cuenta de que necesitamos el concepto de mes y de libro. Por lo tanto optaremos por utilizar tipos estructurados ya que de esta forma será más fácil representar nuestro modelo.

## 2.2. Extensión 1

En la extensión 1 se nos pide realizar un plan de lectura con las siguientes características:

- todos los libros tienen de 0 a N predecesores
- no hay ningún libro paralelo

El objetivo consiste en leer los libros en un orden en el que no se lea un libro antes que su predecesor. Recordemos que para todo libro su predecesor debe leerse en meses anteriores a él.

Este problema es una extensión del prototipo inicial. Debemos generalizar el concepto de predecesor para que un libro pueda tener de 0 a N predecesores. Para plantearnos este problema, no vamos a volver a empezar desde cero sino que intentaremos reutilizar nuestro prototipo anterior.

Observando detalladamente como hemos decidido implementar el nivel básico, nos damos cuenta de que no hace falta modificar el dominio. Haber representado de forma estructurada y general el prototipo anterior nos ha permitido reutilizar completamente el código.

## 2.3. Extensión 2

En la extensión 2 se nos pide, además de incorporar todas las características de la extensión anterior, que puedan haber de 0 a M libros paralelos. Para hacerlo hemos modificado el dominio de la siguiente forma:

La diferencia básica entre esta extensión y la primera, es que, al añadir libros paralelos debemos asegurarnos de que se cumpla la restricción de libros paralelos. Para hacerlo hemos añadido la siguiente sentencia como precondition de pasar\_de\_mes:

```
not (exists (?l - libro)
      (and
        (leyendo_mes_anterior ?l)
        (exists (?p - libro)
          (and
            (paralelo ?l ?p)
            (not (leyendo ?p))
            (not (leyendo_mes_anterior ?p))
            (not (leido ?p))
          )
        )
      )
    )
```

Los libros paralelos tiene la condición de que se tienen de leer en el mismo mes o en un mes de diferencia. Para hacer, que se cumple esta condición, antes de pasar al mes siguiente lo que hacemos es comprobar que si un libro ha estado leído en un mes anterior todos sus libros paralelos tiene de estar leídos o se deben haber leído en este mes, ya que sino no se cumple la condición de libros paralelos.

### 2.3.1. Generador

Hemos creado un generador para hacer juegos de prueba para la extensión 2.

### 2.3.1.1. ¿Cómo se ha hecho?

Primero de todo, cogemos los tres argumentos de entrada, que son los números de libros, el número de libros deseados, el número de libros leídos, número de libros paralelos, tipo de paralelo y la densidad de predecesores. Hay dos tipos de paralelos, el que se indicaría con un 0, es aquel que si por ejemplo quieres 2 libros paralelos, tendrías el libro 0 y 1 que son paralelos y el 2 y 3 que son paralelos. Para el tipo que se indicaría con un 1, es aquel que si te piden 2 libros paralelos el libro 0 y 1 son paralelos y el 1 y el 2 también lo son, podríamos decir que son paralelos encadenados. Y la densidad de predecesores, es una probabilidad entre 0 y 1 de la densidad de aristas del grafo.

Estos datos los utilizamos para crear el fichero del problema, que se llamara:

```
problemagenerador_numerodenodes_numerodeseados_numeroleidos_numeroparalelos_tipoParalelo_
densidadpredecesores.pddl'
```

Posteriormente, creamos un grafo acíclico dirigido random, utilizando la función de `random_graph` y con la probabilidad de densidad “densidadpredecesores” .

```
G=nx.gnp_random_graph(numerodenodes,densidadpredecesores,directed=True)
DAG = nx.DiGraph([(u,v,{'weight':random.randint(1,1)}) for (u,v) in G.edges() if u<v])
```

A continuación, definimos el problema y ponemos los objetos que necesitaremos, para definir los libros, ponemos “libro\_” seguido del número del nodo y le ponemos el tipo “-libro”. También definimos los meses (de enero a diciembre y los dos meses fantasma) y le ponemos el tipo “-mes”.

Posteriormente definimos el estado inicial. Empezamos, por definir cuál mes es el siguiente de cada uno, incluyendo los dos meses fantasma. Después los libros paralelos y los predecesores.

Para definir los paralelos de tipo 0 lo que hacemos es obtener el orden topológico del grafo y definimos los cogemos de dos en dos. Por ejemplo, si el orden topológico es 2, 3, 9, 1, 5, 7 los paralelos serán (2,3) (9,1) (5,7) y quitamos las aristas del grafo que vayan de entre dos paralelos. Para los de tipo 1 hacemos algo muy parecido pero en este caso sería (2,3), (3,9),(9,1),(1,5). En los dos casos tenemos un contador

que cuando llevamos el “numerodeparalelos”, entendiendo como tal que un paralelo es una relación entre dos libros, dejamos de poner más paralelos.

Para los predecesores, consideramos que el nodo A es predecesor del nodo B si hay una arista que va de A a B.

Seguidamente definimos los libros leído, lo que hacemos es volver a generar el orden topológico y cogemos los “numeordeleídos” primeros libros de este orden. El último lo ponemos como “leyendo en el mes anterior”, por si este libro tiene un paralelo que deseamos.

Ahora pondremos los libros deseados. Para hacerlo, cogemos n números aleatoriamente, entre los nodos del grafo, donde n es el número deseado que el usuario ha introducido al principio.

Finalmente definimos los meses que serán fantasma (fin y fin2) y el objetivo, que es que todos los libros que se han deseado estén leídos.

### 2.3.1.2. ¿Cómo ejecutarlo?

Para ejecutarlo, ponemos la ejecución del programas con los argumentos numerodenodos numerodeseados numeroleídos numeroparalelos tipoparalelos densidadpredecesores.

## 2.4. Extensión 3

En esta extensión se nos pedía que los libros tuvieran un número de páginas y que el planificador controlará que en el plan generado no se superarían las 800 páginas al mes. Para hacer esto, hemos utilizado la extensión de fast forward :fluents.

Esta extensión la hemos utilizado de la siguiente forma:

- Creación de funciones:

- Hemos creado 2 funciones:
  - paginas
    - parámetros: libro
    - Imagen: número de páginas del libro
  - paginas\_leidas
    - parámetro: mes
    - Imagen: número de páginas leídas en ese mes
- Comparaciones aritméticas entre funciones
  - ( $\leq$  (+ (paginas ?l) (paginas\_leidas ?m)) 800)
    - En esta comparación, miramos si la suma de las páginas del libro y las leídas en este mes es inferior a 800
- Operación entre funciones
  - (increase (paginas\_leidas ?m) (paginas ?l))
    - Actualizamos la suma

En esta extensión hemos extendido el modelo, ya que hemos creado un apartado de funciones y hemos modificado la precondition de leer, con la comparación aritmética entre funciones explicada anteriormente y su efecto con la operación entre funciones, también explicada anteriormente.

## 3. Juegos de prueba

### 3.1. Nivel básico

En este apartado hablaremos de duología cuando dos libros están ligados por una relación de precedencia.

Cuando usamos el término libro nos referimos a un libro que no pertenece a una duología, es decir, un libro “suelto”. Los nombres de los juegos de prueba son los mismos que se han utilizado en la definición del campo *problem* de nuestros archivos. Para aumentar la legibilidad, hemos copiado en este documento partes de la entrada y la salida. En cualquier caso, se pueden consultar los juegos de prueba en sus respectivos ficheros y también se pueden ejecutar junto con el dominio.

### 3.1.2. Prueba Problem1

Las condiciones de este juego de pruebas son:

- desear leer los dos libros de una duología
- desear leer el segundo libro de una duología sin haber leído el primero ni desearlo
- desear un libro
- haber leído un libro
- añadimos 300 libros con algunas relaciones de predecesor para que el catálogo sea más realista

#### 3.1.1.1. Resultado esperado

Esperamos que el planning resultante nos indique:

- un orden correcto para la lectura de las dos duologías (leer el primero en un mes anterior al segundo)
- leer el libro deseado
- que no incluya el libro que ya hemos leído

#### 3.1.1.2. Entrada simplificada

```
; Libros que ya hemos leído
(leído Juego_de_Tronos)
; Libros que deseamos leer
(deseado Harry_Potter_1)
(deseado Harry_Potter_2)
(deseado Star_Wars_2)
```

(deseado Crepusculo)

#### 3.1.1.4. Salida

step 0: LEER STAR\_WARS\_1 ENERO  
1: LEER CREPUSCULO ENERO  
2: LEER HARRY\_POTTER\_1 ENERO  
3: PASAR\_DE\_MES ENERO FEBRERO  
4: LEER STAR\_WARS\_2 FEBRERO  
5: LEER HARRY\_POTTER\_2 FEBRERO  
6: PASAR\_DE\_MES FEBRERO MARZO  
7: PASAR\_DE\_MES MARZO ABRIL

#### 3.1.1.5. Resultado

El plan obtenido nos indica leer el libro desado y los dos primeros volúmenes de las duologías en enero.

En febrero leemos las segundas partes de ambas duologías.

Por lo tanto podemos afirmar que este plan es correcto.

### 3.1.2. Prueba Problem2

Las condiciones de este juego de pruebas son:

- desear leer el segundo libro de una duología habiendo leído ya el primero
- desear más de un libro
- haber leído más de un libro
- haber leído una duología
- añadimos 300 libros con algunas relaciones de predecesor para que el catálogo sea más realista

#### 3.1.2.1. Resultado esperado

Esperamos que el planning resultante nos indique:

- leer los libros deseados



- no leer el primer libro de la duología
- leer el segundo libro de la duología

### 3.1.2.2. Entrada simplificada

Libros que ya hemos leído:

(leido Harry\_Potter\_1)

(leido Star\_Wars\_1)

(leido Star\_Wars\_2)

(leido l212)

(leido l199)

(leido l123)

Libros que deseamos leer:

(deseado Harry\_Potter\_2)

(deseado Crepusculo)

(deseado Cincuenta\_Sombras\_de\_Grey)

(deseado Juego\_de\_Tronos)

### 3.1.2.3. Salida

step 0: LEER JUEGO\_DE\_TRONOS ENERO  
1: LEER CINCUENTA\_SOMBRAS\_DE\_GREY ENERO  
2: LEER CREPUSCULO ENERO  
3: LEER HARRY\_POTTER\_2 ENERO  
4: PASAR\_DE\_MES ENERO FEBRERO  
5: PASAR\_DE\_MES FEBRERO MARZO

### 3.1.2.4. Resultado

El plan obtenido nos indica leer los libros desados y el segundo volumen de la duología en enero. Esto es adecuado ya que ya hemos leído el primer libro de la duología. En febrero ya no leemos ningún libro porque no hemos necesitado leer una duología completa.

Por lo tanto podemos afirmar que este plan es correcto.

### 3.1.2.5. Conclusión

Hemos construido un planificador que es capaz de retornar un plan que satisface las restricciones de este nivel básico. Cabe remarcar que como los libros solo pueden tener un predecesor como máximo, siempre podremos encontrar un plan que permita leer todos los libros correctamente. Más precisamente, podemos afirmar que se podrán leer todos los libros en como máximo dos meses. En futuras extensiones, con más restricciones, veremos que no siempre es posible encontrar un plan válido.

## 3.2. Extensión 1

En este apartado hablaremos de saga cuando una serie de libros esté relacionada por una relación de predecesor. Hablaremos de primer libro de la saga, segundo libro de la saga y así sucesivamente. Cuando usamos el término libro nos referimos a un libro que no pertenece a una saga, es decir, un libro “suelto”.

Los nombres de los juegos de prueba son los mismos que se han utilizado en la definición del campo *problem* de nuestros archivos.

Para aumentar la legibilidad, hemos copiado en este documento partes de la entrada y la salida. En cualquier caso, se pueden consultar los juegos de prueba en sus respectivos ficheros y también se pueden ejecutar junto con el dominio.

### 3.2.1. Prueba Problem1

Las condiciones de este juego de pruebas son:

- desear leer todos los libros de una saga
- desear leer el quinto libro de una saga sin haber leído ni deseado los cuatro anteriores
- haber leído un libro
- añadimos 300 libros con algunas relaciones de predecesor para que el catálogo sea más realista

### 3.2.1.1. Resultado esperado

Esperamos que el planning resultante nos indique:

- un orden correcto para la lectura de las dos sagas (leer los libros en un orden correcto)
- que no incluya el libro que ya hemos leído

### 3.2.1.2. Entrada simplificada

Libros que ya hemos leído

(leído Crepusculo)

Libros que deseamos leer

(deseado Harry\_Potter\_5)

(deseado Star\_Wars\_1)

(deseado Star\_Wars\_2)

(deseado Star\_Wars\_3)

### 3.2.1.1.3. Salida

*step 0: LEER HARRY\_POTTER\_1 ENERO*  
*1: LEER STAR\_WARS\_1 ENERO*  
*2: PASAR\_DE\_MES ENERO FEBRERO*  
*3: LEER HARRY\_POTTER\_2 FEBRERO*  
*4: LEER STAR\_WARS\_2 FEBRERO*  
*5: PASAR\_DE\_MES FEBRERO MARZO*  
*6: LEER HARRY\_POTTER\_3 MARZO*  
*7: LEER STAR\_WARS\_3 MARZO*  
*8: PASAR\_DE\_MES MARZO ABRIL*  
*9: LEER HARRY\_POTTER\_4 ABRIL*  
*10: PASAR\_DE\_MES ABRIL MAYO*  
*11: LEER HARRY\_POTTER\_5 MAYO*  
*12: PASAR\_DE\_MES MAYO JUNIO*  
*13: PASAR\_DE\_MES JUNIO JULIO*

#### 3.2.1.4. Resultado

El plan obtenido nos indica leer todos los libros de las dos sagas deseadas. Podemos comprobar que se lee una entrega por mes.

Por lo tanto podemos afirmar que este plan es correcto.

#### 3.2.2. Prueba Problem2

Las condiciones de este juego de pruebas son:

- desear leer trece libros de una saga sin haber leído ninguno anteriormente
- haber leído un libro

##### 3.2.2.1. Resultado esperado

Esperamos que el planning resultante nos indique:

- no se ha podido encontrar ningún plan posible

##### 3.2.2.2. Entrada simplificada

Libros que ya hemos leído

(leído Crepusculo)

Libros que deseamos leer

(deseado Harry\_Potter\_13)

##### 3.2.2.3.Salida

*best first search space empty! problem proven unsolvable.*

#### 3.2.2.4. Resultado

Se nos anuncia de que es imposible satisfacer este plan de lectura en un único año.

Este es correcto ya que podemos ver que es imposible leer los 13 libros de la saga si un año tiene únicamente 12 meses. Además, hemos comprobado que efectivamente no se pueden leer libros en los meses fantasma.

#### 3.2.3. Prueba Problem2bis

Podemos realizar una pequeña variación en el ejemplo anterior para comprobar qué pasa si tenemos que leer 12 libros de una misma saga y no 13. Para este propósito, simplemente indicamos que el primer libro de la saga ha sido leído y mantenemos el resto exactamente igual.

#### 3.1.3.1.Salida

*step 0: LEER HARRY\_POTTER\_2 ENERO*  
*1: PASAR\_DE\_MES ENERO FEBRERO*  
*2: LEER HARRY\_POTTER\_3 FEBRERO*  
*3: PASAR\_DE\_MES FEBRERO MARZO*  
*4: LEER HARRY\_POTTER\_4 MARZO*  
*5: PASAR\_DE\_MES MARZO ABRIL*  
*6: LEER HARRY\_POTTER\_5 ABRIL*  
*7: PASAR\_DE\_MES ABRIL MAYO*  
*8: LEER HARRY\_POTTER\_6 MAYO*  
*9: PASAR\_DE\_MES MAYO JUNIO*  
*10: LEER HARRY\_POTTER\_7 JUNIO*  
*11: PASAR\_DE\_MES JUNIO JULIO*  
*12: LEER HARRY\_POTTER\_8 JULIO*  
*13: PASAR\_DE\_MES JULIO AGOSTO*

14: LEER HARRY\_POTTER\_9 AGOSTO  
15: PASAR\_DE\_MES AGOSTO SETIEMBRE  
16: LEER HARRY\_POTTER\_10 SETIEMBRE  
17: PASAR\_DE\_MES SETIEMBRE OCTUBRE  
18: LEER HARRY\_POTTER\_11 OCTUBRE  
19: PASAR\_DE\_MES OCTUBRE NOVIEMBRE  
20: LEER HARRY\_POTTER\_12 NOVIEMBRE  
21: PASAR\_DE\_MES NOVIEMBRE DICIEMBRE  
22: LEER HARRY\_POTTER\_13 DICIEMBRE  
23: PASAR\_DE\_MES DICIEMBRE F1  
24: PASAR\_DE\_MES F1 F2

### 3.2.3.2. Resultado

En este caso sí que existe un plan válido, que corresponde a leer un libro cada mes desde enero hasta diciembre.

### 3.2.5. Prueba Problem3

Las condiciones de este juego de pruebas son:

- desear leer el noveno y el decimotercero libro de una misma saga, habiendo leído los 6 primeros
- desear libros
- haber leído una saga completa

#### 3.2.5.1. Resultado esperado

Esperamos que el planning resultante nos indique:

- un orden correcto para la lectura de la saga de los libros 7 al 13
- que no incluya los libros que ya hemos leído
- que incluya los libros que queremos leer

### 3.2.5.2. Entrada simplificada

Libros que ya hemos leído

(leído Crepusculo)

Libros que deseamos leer

(deseado Harry\_Potter\_5)

(deseado Star\_Wars\_1)

(deseado Star\_Wars\_2)

(deseado Star\_Wars\_3)

### 3.2.5.3. Salida

*step 0: LEER HARRY\_POTTER\_7 ENERO*  
*1: LEER CINCUENTA\_SOMBRAS\_DE\_GREY ENERO*  
*2: LEER JUEGO\_DE\_TRONOS ENERO*  
*3: PASAR\_DE\_MES ENERO FEBRERO*  
*4: LEER HARRY\_POTTER\_8 FEBRERO*  
*5: PASAR\_DE\_MES FEBRERO MARZO*  
*6: LEER HARRY\_POTTER\_9 MARZO*  
*7: PASAR\_DE\_MES MARZO ABRIL*  
*8: LEER HARRY\_POTTER\_10 ABRIL*  
*9: PASAR\_DE\_MES ABRIL MAYO*  
*10: LEER HARRY\_POTTER\_11 MAYO*  
*11: PASAR\_DE\_MES MAYO JUNIO*  
*12: LEER HARRY\_POTTER\_12 JUNIO*  
*13: PASAR\_DE\_MES JUNIO JULIO*  
*14: LEER HARRY\_POTTER\_13 JULIO*  
*15: PASAR\_DE\_MES JULIO AGOSTO*  
*16: PASAR\_DE\_MES AGOSTO SETIEMBRE*

#### 3.2.5.4. Resultado

El plan obtenido nos indica leer todos los libros de la saga del 7 al 13 en un orden correcto. También lee los otros dos libros que queríamos.

Por lo tanto podemos afirmar que este plan es correcto.

### 3.3.Extensión 2

#### 3.3.1. Prueba Ext2-1

Las condiciones de este juego de pruebas son:

- desear leer el libro de Harry Potter 4 y Star Wars 3
- leer el libro 1 de Harry Potter
- tener libros predecesores y paralelos

##### 3.3.1.1. Resultado esperado

Esperamos que el planning resultante nos indique:

- un orden correcto para la lectura
- que no incluya los libros que ya hemos leído
- que incluya los libros que queremos leer

##### 3.3.1.2. Entrada simplificada

Libros que ya hemos leído

(leido Harry\_Potter\_1)

Libros que deseamos leer

deseado Harry\_Potter\_4)

(deseado Star\_Wars\_3)

Paralelos Harry\_Potter\_5 y Star\_Wars\_1



### 3.3.1.3. Salida

*step 0: LEER HARRY\_POTTER\_2 ENERO*  
*1: PASAR\_DE\_MES ENERO FEBRERO*  
*2: LEER HARRY\_POTTER\_3 FEBRERO*  
*3: PASAR\_DE\_MES FEBRERO MARZO*  
*4: LEER STAR\_WARS\_1 MARZO*  
*5: LEER HARRY\_POTTER\_4 MARZO*  
*6: PASAR\_DE\_MES MARZO ABRIL*  
*7: LEER STAR\_WARS\_2 ABRIL*  
*8: LEER HARRY\_POTTER\_5 ABRIL*  
*9: PASAR\_DE\_MES ABRIL MAYO*  
*10: LEER STAR\_WARS\_3 MAYO*  
*11: PASAR\_DE\_MES MAYO JUNIO*  
*12: PASAR\_DE\_MES JUNIO JULIO*

### 3.3.1.4. Resultado

El plan obtenido nos indica leer todos los libros de la saga Harry Potter y Star Wars en un orden correcto hasta los que hemos deseado. Nótese que no puede leerse Star Wars 1 hasta Marzo, porque es paralelo a Harry Potter 5.

Por lo tanto podemos afirmar que este plan es correcto.

### 3.3.2. Prueba Ext2-2

La diferencia con el ejemplo anterior es que hemos añadido dos libros nuevos, y se han puesto como predecesores de starwars\_2 y starwars\_3

### 3.3.2.3. Salida

*step 0: LEER HOLA\_2 ENERO*  
*1: LEER HOLA\_1 ENERO*

2: LEER HARRY\_POTTER\_2 ENERO  
3: PASAR\_DE\_MES ENERO FEBRERO  
4: LEER HARRY\_POTTER\_3 FEBRERO  
5: PASAR\_DE\_MES FEBRERO MARZO  
6: LEER STAR\_WARS\_1 MARZO  
7: LEER HARRY\_POTTER\_4 MARZO  
8: PASAR\_DE\_MES MARZO ABRIL  
9: LEER STAR\_WARS\_2 ABRIL  
10: LEER HARRY\_POTTER\_5 ABRIL  
11: PASAR\_DE\_MES ABRIL MAYO  
12: LEER STAR\_WARS\_3 MAYO  
13: PASAR\_DE\_MES MAYO JUNIO  
14: PASAR\_DE\_MES JUNIO JULIO

#### 3.3.2.4. Resultado

El plan obtenido nos indica leer todos los libros de la saga Harry Potter y Star Wars en un orden correcto hasta los que hemos deseado. Nótese que no puede leerse Star Wars 1 hasta Marzo, porque es paralelo a Harry Potter 5 y nos hace leer los dos libros predecesores antes .

Por lo tanto podemos afirmar que este plan es correcto.

### 3.3.Extensión 3

En esta extensión se nos pedía que los libros tuvieran un número de páginas y que el planificador controlará que en el plan generado no se superarían las 800 páginas al mes. Para hacer esto, hemos utilizado la extensión de pddl :fluents.

Esta extensión la hemos utilizado de la siguiente forma:

- Creación de funciones:
  - Hemos creado 2 funciones:

- paginas
    - parámetros: libro
    - Imagen: número de páginas del libro
  - paginas\_leidas
    - parámetro: mes
    - Imagen: número de páginas leídas en ese mes
- Comparaciones aritméticas entre funciones
  - ( $\leq (+ (\text{paginas } ?l) (\text{paginas\_leidas } ?m)) 800$ )
    - En esta comparación, miramos si la suma de las páginas del libro y las leídas en este mes es inferior a 800
- Operación entre funciones
  - ( $\text{increase} (\text{paginas\_leidas } ?m) (\text{paginas } ?l)$ )
    - Actualizamos la suma

En esta extensión hemos extendido el modelo, ya que hemos creado un apartado de funciones y hemos modificado la precondition de leer, con la comparación aritmética entre funciones explicada anteriormente y su efecto con la operación entre funciones, también explicada anteriormente.

### 3.3.1. Problem1

Las condiciones de este juego de pruebas son:

- libros paralelos
- libros con páginas

Resultado esperado:

Esperamos que el planning resultante nos indique:

- un orden correcto para la lectura de las dos sagas (leer los libros en un orden correcto)
- que los paralelos se lean en 2 meses
- que no se superen las 800 páginas mensuales

Entrada simplificada:

Libros que ya hemos leído

(leído Harry\_Potter\_1)

Libros que deseamos leer

(deseado Harry\_Potter\_4)

(deseado Star\_Wars\_3)

Paralelos Harry\_Potter\_5 y Star\_Wars\_1

Salida:

*step 0: LEER HARRY\_POTTER\_2 ENERO*

*1: PASAR\_DE\_MES ENERO FEBRERO*

*2: LEER HARRY\_POTTER\_3 FEBRERO*

*3: PASAR\_DE\_MES FEBRERO MARZO*

*4: LEER HARRY\_POTTER\_4 MARZO*

*5: PASAR\_DE\_MES MARZO ABRIL*

*6: LEER STAR\_WARS\_1 ABRIL*

7: LEER HARRY\_POTTER\_5 ABRIL  
8: PASAR\_DE\_MES ABRIL MAYO  
9: LEER STAR\_WARS\_2 MAYO  
10: PASAR\_DE\_MES MAYO JUNIO  
11: LEER STAR\_WARS\_3 JUNIO  
12: PASAR\_DE\_MES JUNIO JULIO  
13: PASAR\_DE\_MES JULIO AGOSTO

Resultado:

El plan obtenido nos indica leer todos los libros de las dos sagas deseadas. Podemos comprobar que no se supera el número de páginas en ningún mes. Los libros de las dos sagas se leen en un orden correcto y los libros paralelos se leen ambos en abril. Nótese que de no ser porque el número de páginas de Harry\_Potter\_4 y Star\_Wars\_1 suman más de 800 Star\_Wars\_1 se leería el mes anterior, tal como ocurre en la prueba Ext2-1. Por lo tanto podemos afirmar que este plan es correcto.

### 3.3.2. Problem2

Las condiciones de este juego de pruebas son:

- leer un libro con más de 800 páginas
- leer otros libros de menos de 800 páginas

Resultado esperado:

Esperamos que el planning resultante nos indique:

- no se ha podido encontrar ningún plan posible

Entrada simplificada:

Libros que ya hemos leído

(leído Crepusculo)

Libros que deseamos leer

(deseado Harry\_Potter\_13)

Salida:

*best first search space empty! problem proven unsolvable.*

Resultado:

Se nos anuncia de que es imposible satisfacer este plan de lectura en un único año.

Este es correcto ya que podemos ver que es imposible leer un libro de más de 800 páginas, en ningún mes.