

# **Intel·ligència Artificial: Pràctica Planificació**

*Guillem Cabré, Carla Cordero, Hannah Röber*

Curs 2024-25, Quadrimestre de tardor

# Continguts

<b>1</b>	<b>Introducció</b>	<b>2</b>
<b>2</b>	<b>Modelatge del Domini</b>	<b>3</b>
2.1	Variables . . . . .	3
2.2	Predicats . . . . .	3
2.3	Funcions . . . . .	4
2.4	Accions . . . . .	4
<b>3</b>	<b>Modelatge dels Problemes</b>	<b>7</b>
3.1	Objectes . . . . .	7
3.2	Estat Inicial . . . . .	7
3.3	Estat Final . . . . .	7
<b>4</b>	<b>Desenvolupament del Model</b>	<b>8</b>
4.1	Nivell Bàsic . . . . .	8
4.2	Extensió 1 . . . . .	8
4.3	Extensió 2 . . . . .	8
4.4	Extensió 3 . . . . .	8
4.5	Extensió 4 . . . . .	8
<b>5</b>	<b>Jocs de prova</b>	<b>9</b>
<b>6</b>	<b>Conclusió</b>	<b>10</b>

# 1 Introducció

El problema plantejat consisteix en la creació d'una eina capaç de planificar el visionat de continguts audiovisuals per a un usuari, tenint en compte diversos factors com el catàleg de continguts disponibles, les dependències entre aquests, els continguts ja visionats i aquells que es volen veure. L'objectiu és generar un pla que distribueixi els continguts al llarg dels dies, complint amb restriccions com el respecte als precedents i continguts paral·lels.

La pràctica s'ha de desenvolupar amb el planificador *Fast Forward* v2.3 i inclou diverses extensions que permeten incrementar la dificultat i el valor de l'activitat. Es posa especial èmfasi en la correcta modelització del domini i dels problemes, així com en la qualitat dels jocs de prova utilitzats. Finalment, la documentació i els resultats obtinguts seran avaluats en funció de la seva qualitat i adequació.

La pràctica es centra en resoldre un problema de planificació utilitzant el llenguatge de descripció PDDL. El nivell bàsic del problema implica la generació d'un pla de visionat on cada contingut pot tenir com a màxim un contingut predecessor, i no existeixen continguts paral·lels. El planificador ha de garantir que els continguts es visionen en l'ordre correcte, respectant aquestes dependències senzilles.

Les extensions incrementen la complexitat del problema afegint noves restriccions i funcionalitats:

- **Extensió 1:** Permet que un contingut tingui múltiples predecessors. El planificador ha de garantir que tots els predecessors es visionin abans del contingut corresponent.
- **Extensió 2:** Incorpora continguts paral·lels, que es poden visionar el mateix dia o en dies consecutius. Això reflecteix les relacions complexes entre subtrames o universos compartits.
- **Extensió 3:** Limita el nombre màxim de continguts a tres per dia, afegint restriccions al nombre d'activitats diàries.
- **Extensió 4:** Afegeix una durada en minuts als continguts i estableix un límit màxim de 200 minuts per dia, requerint un control més precís dels recursos temporals.

Aquestes extensions ofereixen als estudiants la possibilitat d'abordar reptes més avançats i de demostrar una comprensió més profunda de la planificació en espai d'estats, així com de millorar el seu domini del llenguatge PDDL. Cada extensió també es tradueix en un increment potencial de la nota màxima de la pràctica, incentivant el desenvolupament de solucions més completes i sofisticades.

## 2 Modelatge del Domini

El domini **REDFLIX** s'ha creat per capturar les dependències entre continguts audiovisuals (pel·lícules, capítols de sèries) i les restriccions associades amb la seva assignació a dies en un pla de visionat.

### 2.1 Variables

Les variables que representen el nostre domini son dues. Aquestes són:

- **content**: Representa els continguts audiovisuals (capítols o pel·lícules).
- **day**: Representa els dies disponibles per al pla de visionat.

### 2.2 Predicats

Els predicats representen les relacions i restriccions entre els continguts i els dies i són la base per definir les precondicions i efectes de les accions. Mostrarem primer els predicats del problema amb **Nivell senzill**. I a continuació afegirem els predicats conforme avancem per les diferents extensions.

- (**watched** ?c - content): Indica que un contingut ?c ja s'ha vist.
- (**is\_wanted** ?c - content): Indica que l'usuari pot veure un contingut ?c.
- (**predecessor** ?c1 - content ?c2 - content): Indica que el contingut ?c1 és predecessor de ?c2. Per tant, ?c1 s'ha de veure abans que ?c2.
- (**day\_to\_watch** ?c - content ?d - day): Assigna un contingut ?c a un dia ?d.
- (**yesterday** ?d1 - day ?d2 - day): Indica que ?d1 és el dia anterior a ?d2.
- (**assigned** ?c - content): Marca que un contingut ?c ja té un dia assignat.

Siguin aquests els predicats del **Nivell bàsic**, ara enumerarem els predicats de les altres extensions:

**Extensió 1:** Els predicats romanen iguals, l'únic que canviarà serà la post condició quan s'assigni un contingut a un dia.

**Extensió 2:** S'afegeix el predicat de contingut paral·lel (**parallel**).

- (**parallel** ?c1 - content ?c2 - content): Estableix que ?c1 és paral·lel amb ?c2 i s'han de veure el mateix dia o en dies consecutius. Hem assumit la relació paral·lel és unidireccional, però el nostre model permet que sigui bidireccional, si en el problema s'escriuen:  
(parallel A B)  
(parallel B A)

**Extensió 3:** Apareixen tres predicats nous, cada un indicant quin dels tres continguts de cada dia han sigut ja assignats. Aquestes són:

- (**assigned\_one** ?d - day): S'ha assignat el primer contingut el dia ?d.
- (**assigned\_two** ?d - day): S'ha assignat el segon contingut el dia ?d.
- (**assigned\_three** ?d - day): S'ha assignat el tercer contingut el dia ?d.

**Extensió 4:** Els tres predicats de la extensió anterior són retirats. Tampoc s'afegeix cap predicat nou en aquesta extensió. En aquesta extensió es farà ús de fluents, s'explicarà a continuació.

## 2.3 Funcions

Les funcions introdueixen fluents per gestionar informació numèrica. Aquestes només es fan servir per l'Extensió 4:

- (total-days): Enregistra el nombre total de dies utilitzats. Necessari per optimitzar el pla.
- (duration ?c - content): Assigna la durada de cada contingut, modelant restriccions de temps.
- (day\_duration ?d - day): Conté la durada acumulada dels continguts assignats a un dia.
- (remaining-content): Indica el nombre de continguts desitjats encara no assignats, permetent controlar l'avanç cap a l'objectiu.

## 2.4 Accions

En totes les extensions hem fet ús de tres accions. Per explicar-les farem una breu desprescripció del seu funcionament i explicarem les diferents iteracions que han tingut al llarg de les diferents extensions. Per simplificar, direm que tenim dos planificadors diferents, el de la **Extensió 3** i el de la **Extensió 4**, si les accions d'aquests dos planificadors son diferents, analitzarem en que canvien.

Les 3 accions són les següents:

### add\_content

Aquesta acció afegeix continguts que l'usuari vol veure o que el sistema vol que l'usuari vegi.

- **Precondicions:** El contingut no s'ha vist i és desitjat (`is_wanted`).
- **Efectes:** Inclou els predecessors o continguts paral·lels a la llista de continguts desitjats.

---

```
1 (:action add_content
2   :parameters (?c - content)
3   :precondition (and
4     (is_wanted ?c)
5     (not (watched ?c)))
6   :effect (and
7     ;; Si el contingut vist te un successor o algun contingut en parallel per
7     veure i no ha estat vist, s'afegeix a la llista de continguts per veure
8     (forall (?c2 - content)
9       (when (and (or (predecessor ?c2 ?c) (parallel ?c2 ?c)) (not (watched
10         ?c2))))
10       (is_wanted ?c2))
11   )))
```

---

Listing 1: Acció add\_content

### set\_day\_unique

Assigna un dia a un contingut que no té successors.

- **Precondicions:** El contingut és desitjat, no s'ha vist, no té successors ni paral·lels sense assignar, i el dia té espai disponible.
- **Efectes:** Marca el contingut com a assignat i reserva el dia corresponent.

---

```
1 (:action set_day_unique
2   :parameters (?c - content ?d - day)
3   :precondition (and
4     (is_wanted ?c)
5     (not (watched ?c))
6     (or
7       (not (assigned_one ?d))
```

```

8      (not (assigned_two ?d))
9      (not (assigned_three ?d)))
10    ;; Verifica que no haya sucesores que no hayan sido vistos
11    (not (exists (?c2 - content)
12      (and
13        (predecessor ?c ?c2)
14        (not (watched ?c2))))))
15    ;; Verifica que no haya contenidos paralelos que no hayan sido vistos
16    (not (exists (?c2 - content)
17      (and
18        (parallel ?c ?c2)
19        (not (watched ?c2))))))
20    :effect (and
21      (day_to_watch ?c ?d)
22      (when (not (assigned_one ?d)) (assigned_one ?d))
23      (when (and (assigned_one ?d) (not (assigned_two ?d))) (assigned_two ?d))
24      (when (and (assigned_one ?d) (assigned_two ?d) (not (assigned_three ?d)))
25        (assigned_three ?d))
26      (assigned ?c)
27    ))

```

Listing 2: Acció set\_day-unique

Per l'**Extensió 2** s'afegeixen les línies que verifiquen que no hi hagi continguts paral·lels que no hagin sigut vistos. Després per l'**Extensió 3**, s'afegeixen les línies de la precondició or de (not (assigned\_<number> ?d)), i les de la postcondició també.

Per altra banda, l'**Extensió 4** elimina els canvis fets per la tercera extensió i afegeix a la precondició: ( $\leq$  (+ (day\_duration ?d) (duration ?c)) 200): s'assegura que el dia ?d no se li assignin més de 200 minuts.

També es modifica la postcondició, a la qual se li afegeixen les tres següents línies:

(increase (day\_duration ?d) (duration ?c)): incrementa en duració de continguts visionats en el dia ?d en duració de ?c minuts de visionat.  
 (increase (total-days) 1): incrementa els dies usats.  
 (decrease (remaining-content) 1): decrementa els continguts per veure

## set\_day

Assigna un dia a un contingut que pot tenir predecessors o paral·lels.

- **Precondicions:** El contingut és desitjat, compleix amb les relacions de predecessors i paral·lels, i el dia té espai disponible.
- **Efectes:** Assigna el contingut al dia seleccionat, respectant les restriccions.

```

1  (:action set_day
2    :parameters (?c1 ?c2 - content ?d1 ?d2 - day)
3    :precondition (and
4      (is_wanted ?c1)
5      (day_to_watch ?c2 ?d2)
6      (or
7        (not (assigned_one ?d1))
8        (not (assigned_two ?d1))
9        (not (assigned_three ?d1)))
10     (or
11       ;condiciones para contenido paralelo
12       (and
13         (parallel ?c1 ?c2)
14         (or
15           (previous ?d1 ?d2)
16           (= ?d1 ?d2)))
17       ;condiciones para contenido predecessor

```

```

18         (and
19           (predecessor ?c1 ?c2)
20           (previous ?d1 ?d2))))
21 :effect (and
22   (day_to_watch ?c1 ?d1)
23   (when (not (assigned_one ?d1)) (assigned_one ?d1))
24   (when (and (assigned_one ?d1) (not (assigned_two ?d1))) (assigned_two
    ?d1))
25   (when (and (assigned_one ?d1) (assigned_two ?d1) (not (assigned_three
    ?d1))) (assigned_three ?d1))
26   (assigned ?c1)
27 ))

```

---

Listing 3: Acció set\_day

Per l'**Extensió 2** s'afegeixen les línies amb la condició dels continguts paral·lels. Després per l'**Extensió 3**, s'afegeixen les línies de la precondició or de (not (assigned-<number> ?d)), i les de la postcondició també.

Els canvis de l'**Extensió 4** son idèntics als de l'acció anterior: eliminar els canvis de l'extensió 3 i afegir les línies a la precondició i postcondició:

(<= (+ (day\_duration ?d) (duration ?c)) 200): s'assegura que el dia ?d no se li assignin més de 200 minuts.

(increase (day\_duration ?d) (duration ?c)): incrementa en duració de continguts visionats en el dia ?d en duració de ?c minuts de visionat.

(increase (total-days) 1): incrementa els dies usats.

(decrease (remaining-content) 1): decrementa els continguts per veure

## 3 Modelatge dels Problemes

Els problemes específics es modelen com a instàncies del domini **Redflix**, definint els objectes, l'estat inicial i l'estat final.

### 3.1 Objectes

Es defineixen els objectes com a continguts i dies. Per exemple:

```
(:objects
  bb_s1 bb_s2 bb_s3 cc1 cc2 - content
  day1 day2 day3 day4 - day
)
```

### 3.2 Estat Inicial

L'estat inicial descriu les relacions entre continguts i dies, incloent-hi:

- Continguts ja vistos.
- Continguts que l'usuari vol veure.
- Relacions de predecessors i paral·lels.
- Relacions temporals entre els dies.

Exemple:

```
(:init
  (is_wanted bb_s3)
  (watched bb_s1)
  (watched bb_s2)
  (predecessor bb_s1 bb_s2)
  (predecessor bb_s2 bb_s3)
  (parallel cc1 cc2)
  (yesterday day1 day2)
  (yesterday day2 day3)
)
```

### 3.3 Estat Final

L'estat objectiu especifica que tots els continguts desitjats han de ser assignats a dies, respectant les relacions de predecessors i paral·lels. Per exemple:

```
(:goal
  (and
    (day_to_watch bb_s3 day3)
    (day_to_watch cc1 day2)
    (day_to_watch cc2 day2)
  )
)
```



## 4 Desenvolupament del Model

El model s'ha desenvolupat iterativament, abordant les extensions de manera progressiva:

### 4.1 Nivell Bàsic

Es va començar modelant continguts amb com a màxim un predecessor. Les accions no consideraven continguts paral·lels ni límits diaris.

### 4.2 Extensió 1

Es van afegir múltiples predecessors al model. Es van validar les accions per assegurar que les dependències es respectaven.

### 4.3 Extensió 2

Es van incorporar continguts paral·lels, garantint que es veien el mateix dia o en dies consecutius.

### 4.4 Extensió 3

Es van afegir restriccions per limitar el nombre de continguts per dia a tres. Això es va aconseguir amb els predicats `assigned_one`, `assigned_two` i `assigned_three`.

### 4.5 Extensió 4

\*\*\*\*\* "Una breve explicación de como habéis desarrollado los modelos (de una sola vez, por iteraciones)" —¿, no sé exactament què dir

## 5 Jocs de prova

\*\*\*\*\*El conjunto de problemas de prueba (mínimo 2 por extensión), 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. Si habéis implementado el generador de problemas, al menos uno de los juegos de prueba de cada extensión ha de ser obtenido de este.

## 6 Conclusió