

Intel·ligència Artificial: Pràctica Planificació

Guillem Cabré, Carla Cordero, Hannah Röber

Curs 2024-25, Quadrimestre de tardor

Continguts

1	Modelatge del Domini	2
1.1	Variables	2
1.2	Predicats	2
1.3	Funcions	2
1.4	Accions	2
1.4.1	add_content	2
1.4.2	set_day_unique	3
1.4.3	set_day	3
2	Modelatge dels Problemes	3
2.1	Objectes	3
2.2	Estat Inicial	3
2.3	Estat Final	4
3	Desenvolupament del Model	4
3.1	Nivell Bàsic	4
3.2	Extensió 1	4
3.3	Extensió 2	4
3.4	Extensió 3	4
3.5	Extensió 4	4
4	Jocs de prova	4
5	Conclusió	4

1 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.

1.1 Variables

S'han definit dos tipus principals:

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

1.2 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. Són els següents:

- **(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 ?c2 - content ?c1 - content)**: Defineix que ?c1 s'ha de veure abans que ?c2.
- **(day_to_watch ?c - content ?d - day)**: Assigna un contingut ?c a un dia ?d.
- **(previous ?d1 - day ?d2 - day)**: Indica que ?d1 és el dia anterior a ?d2.
- **(parallel ?c1 - content ?c2 - content)**: Estableix que ?c1 i ?c2 són paral·lels i s'han de veure el mateix dia o en dies consecutius.
- **(assigned ?c - content)**: Marca que un contingut ?c ja té un dia assignat.
- **(assigned_one ?d - day), (assigned_two ?d - day), (assigned_three ?d - day)**: Indiquen que s'han assignat un, dos o tres continguts a un dia ?d.

1.3 Funcions

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

- **(total-days)**: Registra 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)**: Registra 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.

1.4 Accions

S'han definit tres accions principals per construir el pla de visionat:

1.4.1 add_content

Aquesta acció afegeix continguts relacionats a la llista de continguts que l'usuari vol veure.

- **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.

1.4.2 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.

1.4.3 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.

2 Modelatge dels Problemes

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

2.1 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
)
```

2.2 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)
(previous day1 day2)
(previous day2 day3)
)
```

2.3 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)
)
)
```

3 Desenvolupament del Model

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

3.1 Nivell Bàsic

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

3.2 Extensió 1

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

3.3 Extensió 2

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

3.4 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`.

3.5 Extensió 4

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

4 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.

5 Conclusió