ABIA - PDDL: PLANIFICACIÓ

Nils Duran, Ramon Andreu

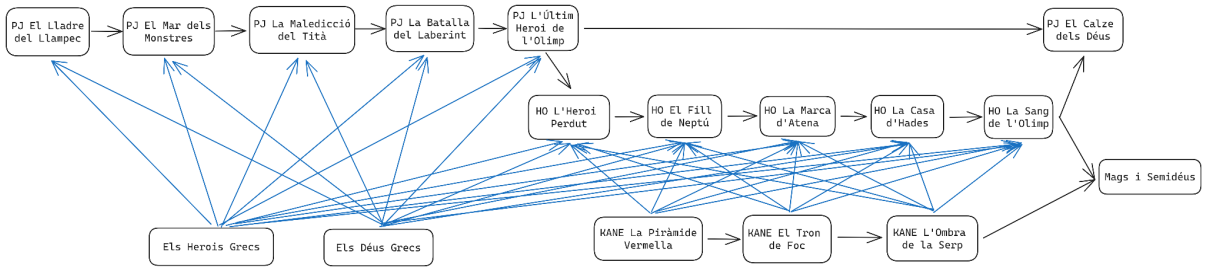
Novembre - Desembre del 2023

Índex

Motivació

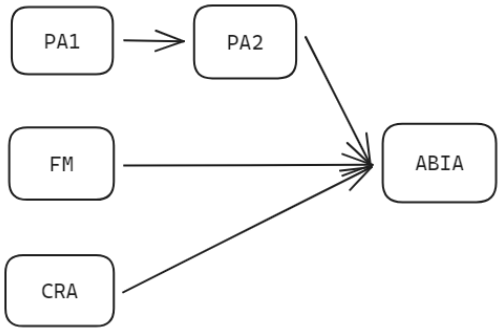
Aquest setembre va sortir un llibre nou de la saga de Percy Jackson. No recordava la sèrie original, que havia llegit fa vuit anys, i volia rellegir els llibres necessaris, però n'hi havia molts altres que havien sortit des d'aleshores, amb relacions entre ells complexos. Evidentment, no vaig fer servir el PDDL per a decidir quins llibres havia de llegir i quins no calia, però hem utilitzat aquest cas real com a base dels nostres fitxers (amb alguns altres llibres i sèries), i les extensions més avançades sí que servirien per a crear un pla vàlid de lectura.

Hem fet un esquema perquè sigui més fàcil entendre el problema. Aquí tenim el subconjunt de la saga de Percy Jackson. Les sagetes negres representen predecessors i les blaves paral·lels. No hem fet servir tots els llibres per a tots els dominis, els hem anat afegint quan tocava.



L'univers de Harry Potter no és tan enrevessat:

Per acabar vam agafar inspiració dels requisits de l'assignatura d'ABIA, segons la guia docent (CRA, FM, PA1 i PA2).



Primer Domini (Nivell Bàsic):

En aquest primer domini se'ns demana el següent:

Nivell bàsic: *Al pla de lectura tots els llibres tenen 0 o 1 predecessors i cap paral·lel. El planner és capaç de trobar un pla per poder arribar a llegir els llibres objectiu encadenant llibres, on cada llibre només té un o cap predecessor.*

Aquest primer planner serà bastant fàcil de fer.
Comencem des de l'inici, **quins requisits ens faran falta?**
(el requisit *stripes* és un bàsic indispensable)

Com ja sabem tots els objectes seran llibres, per tant, a primera vista no farà falta utilitzar el requisit *typing*. Encara que podríem separar els llibres per Saga/Univers, això significaria que el nostre domini estaria molt personalitzat segons els llibres que hi hagi, i perdrem el funcionament universal del domini. No usarem el requisit *typing*.

El següent requisit que podem mirar d'implementar és el *fluents*. A causa de la baixa complexitat del problema no creiem que sigui necessari la implementació de fluents. En tot cas, mentre que l'anem construint a poc a poc, veurem si fa falta, o no.

Finalment, tenim el requisit *adl*. Aquest sí que l'implementarem ja que ens pot ser útil a l'hora de consultar precondicions o aplicar efectes.

Ara que ja tenim els nostres requisits, **quins predicats seran útils?**

Nosaltres hem optat per utilitzar només tres predicats:

(delCataleg ?ll): Com demana a l'enunciat, el planner ha de tenir coneixement sobre els llibres del catàleg.

(predecessor ?ll1 ?ll2): A l'enunciat també es demana que el planner ha de tindre coneixement sobre els llibres predecessors a un altre llibre. El predicat funciona de la següent manera: Segons la nostra implementació, el llibre *?ll1* serà predecessor de *?ll2*.

(llegit ?ll): Aquest predicat serveix per marcar els llibres que ja ha llegit, com es demana a l'enunciat.

L'enunciat també ens demana que el planificador ha de saber, quins llibres vol llegir l'usuari, per complir això, intentarem estalviar-nos un predicat. Més endavant l'implementarem per optimitzar i guiar el planner, però en un domini més avançat.
Per acabar el domini, **ens faltarà acabar les accions**:

Nosaltres hem aconseguit que el planner sigui funcional amb una sola acció. Aquesta acció l'hem anomenada *llegir_llibre.*

El nom és bastant explícit, simplement llegirà els llibres. A més de la manera en què està fet, no només servirà per llibres que tinguin entre zero i un predecessor, sinó que servirà per llibres amb zero o N predecessors.

Com que l'únic paràmetre serà un llibre, el que volem llegir.

La precondició és més complexa:

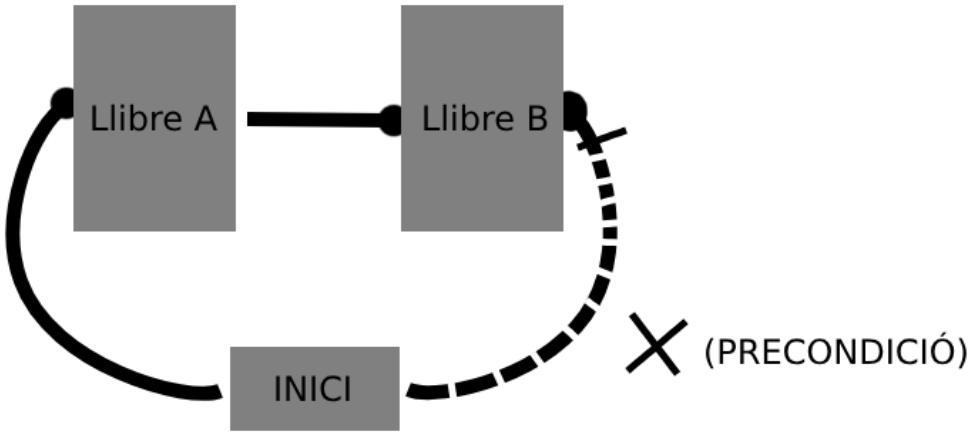
```
:precondition (and (delCataleg ?ll)
(not (exists (?p) (and (predecessor ?p ?ll) (not (llegit ?p))))))
```

Per començar, tenim la condició més restrictiva, que sigui del catàleg. Si el llibre no és del catàleg, la precondició serà falsa ràpidament i el nostre planner serà més eficient.

Després tenim l'última condició, que perquè s'entengui més fàcilment, serà millor traduir-la a llenguatge natural:

"No existeix cap llibre ?p tal que ?p és predecessor de ?ll i ?p encara no s'ha llegit."

Si l'objectiu és el Llibre B, i el Llibre A és predecessor del Llibre B passarà el següent: Intentarà llegir el Llibre B però la precondició s'ho impedeix ja que aquest té un predecessor no llegit. Per tant llegirà el Llibre A, i en conseqüència podrà llegir el Llibre B.



Per acabar, tenim que l'efecte sobre el objecte paràmetre és que marca amb el predicat (llegit ?ll), el llibre que ha entrat com a paràmetre.
El domini ja està acabat, ara només faltarà crear el nostre arxiu problema:
Tenim dues sagues, Percy Jackson i Harry Potter:

```
HP_I_La_Pedra_Filosofal             HP_I_La_Cambra_Secreta
HP_I_El_Pres_De_Azkaban             HP_I_El_Calze_De_foc
HP_I_La_Ordre_Del_Fenix             HP_I_El_Misteri_Del_Princep
HP_I_Les_Reliquies_De_La_Mort

PJ_El_Lladre_Del_Llampec            PJ_El_Mar_Dels_Monstres
```

```
PJ_La_Malediccio_Del_Tita                    PJ_La_Batalla_Del_Laberint
PJ_L_Ultim_Heroi_De_L_Olimp                  PJ_El_Calze_Dels_Deus
```

Una vegada definits com a objectes, cal fer el init:
Primer s'han de definir tots el llibres que vulguem al catàleg, dos de cada saga, per exemple,

```
(delCataleg HP_I_La_Pedra_Filosofal)
(delCataleg HP_I_La_Cambra_Secreta)
(delCataleg PJ_El_Lladre_Del_Llampec)
(delcataleg PJ_El_Mar_Dels_Monstres)
```

Després els predecessors,

```
(predecessor HP_I_La_Pedra_Filosofal HP_I_La_Cambra_Secreta)
(predecessor PJ_El_Lladre_Del_Llampec PJ_El_Mar_Dels_Monstres)
```

Init acabat, ara toca només fer el goal i el nostre problema ja estaria llest per funcionar:
Aquí podem posar que llegeixi les dues sagues:

```
(llegit HP_I_La_Cambra_Secreta)
(llegit PJ_El_Mar_Dels_Monstres)
```

O només una,

```
(llegit PJ_El_Mar_Dels_Monstres)
```

En el nostre fitxer estan definits com a predecessors tots el llibres de les dues sagues, per si és vol provar amb varies parelles o grups més grans. Per exemple, si només volem llegir els dos últims llibres de cada saga, caldrà treure la resta de llibres del catàleg i posar al goal l'últim llibre de cada saga.

Hi ha dos problemes, *problema0_0.pddl*, el qual només llegeix d'una saga (HP), i el *problema0_1.pddl*, que llegeix les dues sagues. Així veiem si en el primer cas, que és el més bàsic, ho fa bé, i en el segon mirem que no hi hagin problemes entre sagues.

Resultats del problema0_0: (metricff.exe -o domini0.pddl -f problema0_0.pddl)

```
0: LLEGIR_LLIBRE HP_I_LA_PEDRA_FILOSOFAL
1: LLEGIR_LLIBRE HP_I_LA_CAMBRA_SECRETA
2: LLEGIR_LLIBRE HP_I_EL_PRES_DE_AZKABAN
3: LLEGIR_LLIBRE HP_I_EL_CALZE_DE_FOC
4: LLEGIR_LLIBRE HP_I_LA_ORDRE_DEL_FENIX
5: LLEGIR_LLIBRE HP_I_EL_MISTERI_DEL_PRINCEP
6: LLEGIR_LLIBRE HP_I_LES_RELIQUIES_DE_LA_MORT
```

Podem veure que el planner llegeix els llibres de forma correcta i ordenada, al ser un nivell fàcil, no hi ha gaire a comentar.

Resultats del problema0_1: (metricff.exe -o domini0.pddl -f problema0_1.pddl)

```
0: LLEGIR_LLIBRE HP_I_LA_PEDRA_FILOSOFAL
1: LLEGIR_LLIBRE PJ_EL_LLADRE_DEL_LLAMPEC
```

```
2: LLEGIR_LLIBRE HP_I_LA_CAMBRA_SECRETA
3: LLEGIR_LLIBRE PJ_EL_MAR_DELS_MONSTRES
4: LLEGIR_LLIBRE HP_I_EL_PRES_DE_AZKABAN
5: LLEGIR_LLIBRE PJ_LA_MALEDICCIO_DEL_TITA
6: LLEGIR_LLIBRE HP_I_EL_CALZE_DE_FOC
7: LLEGIR_LLIBRE PJ_LA_BATALLA_DEL_LABERINT
8: LLEGIR_LLIBRE HP_I_LA_ORDRE_DEL_FENIX
9: LLEGIR_LLIBRE PJ_L_ULTIM_HEROI_DE_L_OLIMP
10: LLEGIR_LLIBRE HP_I_EL_MISTERI_DEL_PRINCEP
11: LLEGIR_LLIBRE PJ_EL_CALZE_DELS_DEUS
12: LLEGIR_LLIBRE HP_I_LES_RELIQUIES_DE_LA_MORT
```

Aquí podem veure que, encara que hi hagi dues sagues, els resultats segueixen sent coherents i correctes.

Segon Domini (Extensió 1):

Què ens demanen per aquest nivel?

Extensió 1: *Els llibres poden tenir de 0 a N predecessors però cap paral·lel. El planner és capaç de construir un pla per poder arribar a llegir els llibres objectiu, on per a tot llibre que pertany al pla, tots els seus llibres predecessors pertanyen al pla i estan en mesos anteriors.*

Aquí ja comença a complicar-se la cosa, però aquí ja tenim una mica fet, ja que el nivell anterior està fet per ser capaç de funcionar amb llibres que tenen de zero a N predecessors. Però primer comencem amb el domini:

Quins requisits ens faran falta?
(el requisit *stripes* és un basic indispensable)

Si pensem en posar el requisit *typing*, passa el mateix que en l'anterior nivel, i per tant no ho farem.

Ara anem amb el requisit *fluents*. Ara sí que la utilitzarem pel següent motiu, utilitzarem un fluent com a variable booleana, per així poder saber quan canviar de més i poder guiar al nostre planificador.

Com utilitzarem de base al planner anterior, necessitarem obligatòriament el requisit *adl*.

Definim el nostre fluent "booleà":

(MesSeguent), ens indicarà quan canviar de més.

El següent pas son els predicats, fan falta més?

En el nostre cas sí però quan definim les noves accions s'explicarà l'ús d'aquests.

(mes_anterior ?ll), (mes_anterior2 ?ll),

Sí, els objectes en aquest predicat son llibres, i marcaran el llibres que s'han llegit el més anterior al actual.

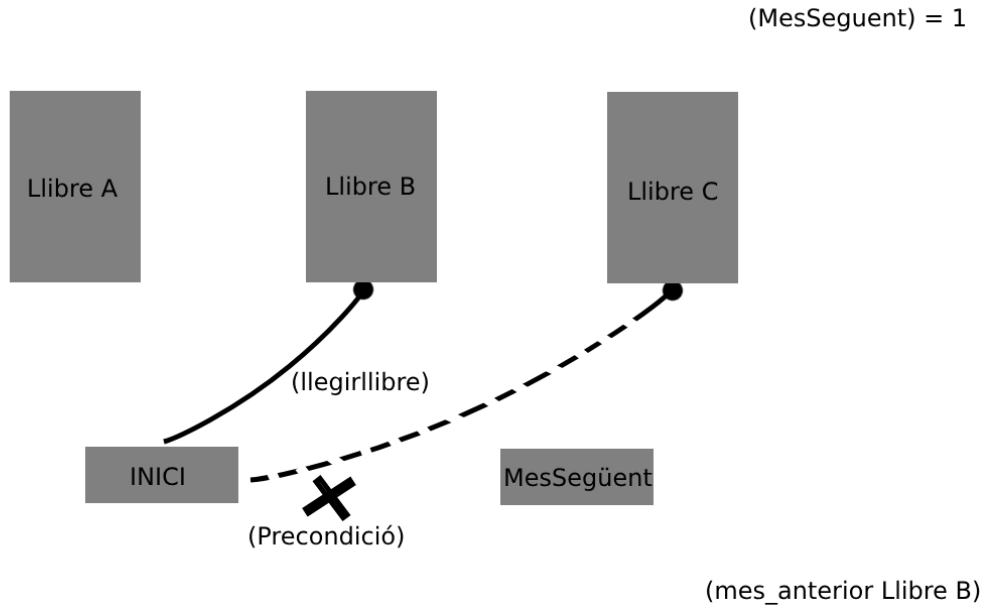
Ara passem directament a les accions, que són el que fan aquest últim pas per acabar la primera extensió:

Tenim tres accions, *llegir_llibre* (amb unes petites modificacions), *llegir_llibre_auxiliar* i *Següent_Mes*. Primer explicarem la mecánica de funcionament i després entrarem en detall en les accions.

Inicialment només es podran executar les accions *llegir_llibre* i *llegir_llibre_auxiliar*, i ho diem així perquè el nostre pla és que primer s'executi l'acció *llegir_llibre*, però al planificador li dona una mica igual.

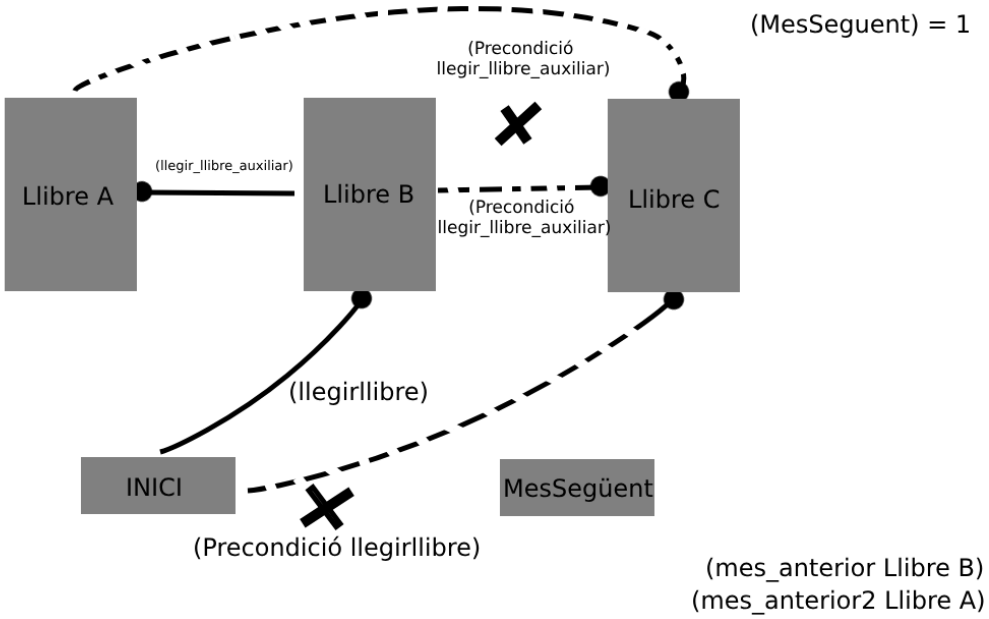
Per que quedi més clar, explicarem amb un exemple el que passa a partir d'aquí:

Si volem llegir el Llibre C, però aquest té com a predecessors el Llibre A i el Llibre B, passarà el següent: Intentarà llegir el Llibre C, no podrà, i per tant intentarà llegir el Llibre B, per exemple. Per això ho farà amb *llegir_llibre*. Els efectes seran els següents: marca com llegit el Llibre B, canvia la variable *MesSegüent* a 1 i per últim marca al Llibre B amb el predicat *(mes_anterior Llibre B)*.

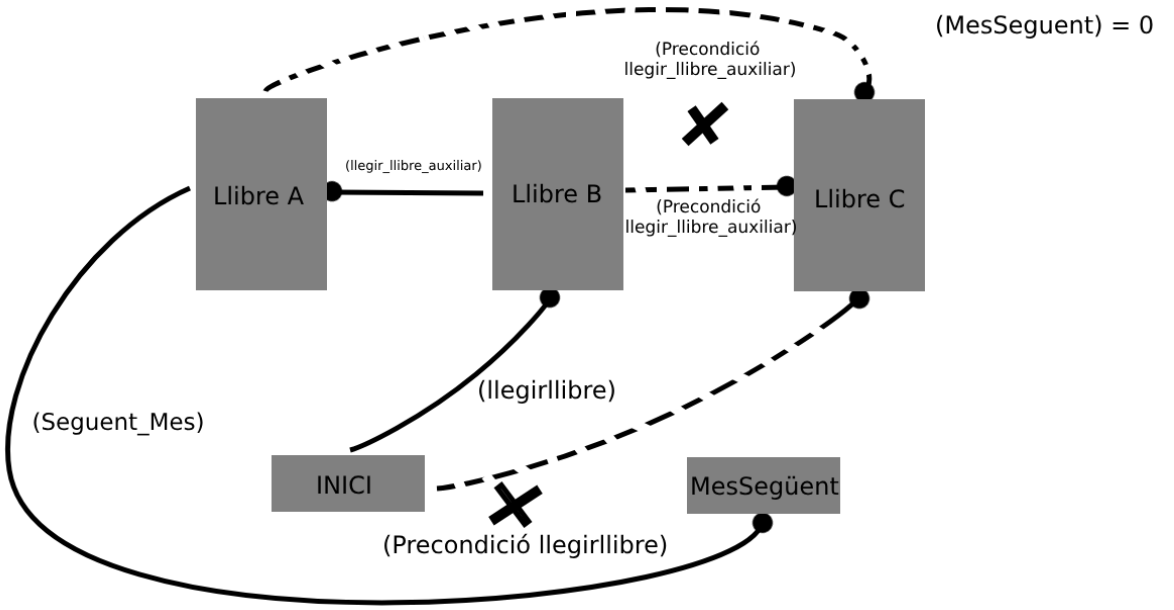


I què significa això? Que ara, el planificador té l'opció de canviar de mes, però ja no podrà tornar a utilitzar l'acció *llegir_llibre*. Aquí és on entra en joc l'acció *llegir_llibre_auxiliar*, que llegirà tots els llibres que no tinguin un predecessor que encara no s'ha llegit i que aquest predecessor no tingui la marca de *(mes_anterior ?ll)* o *(mes_anterior2 ?ll)*. (A més també mirarà si és del catàleg).

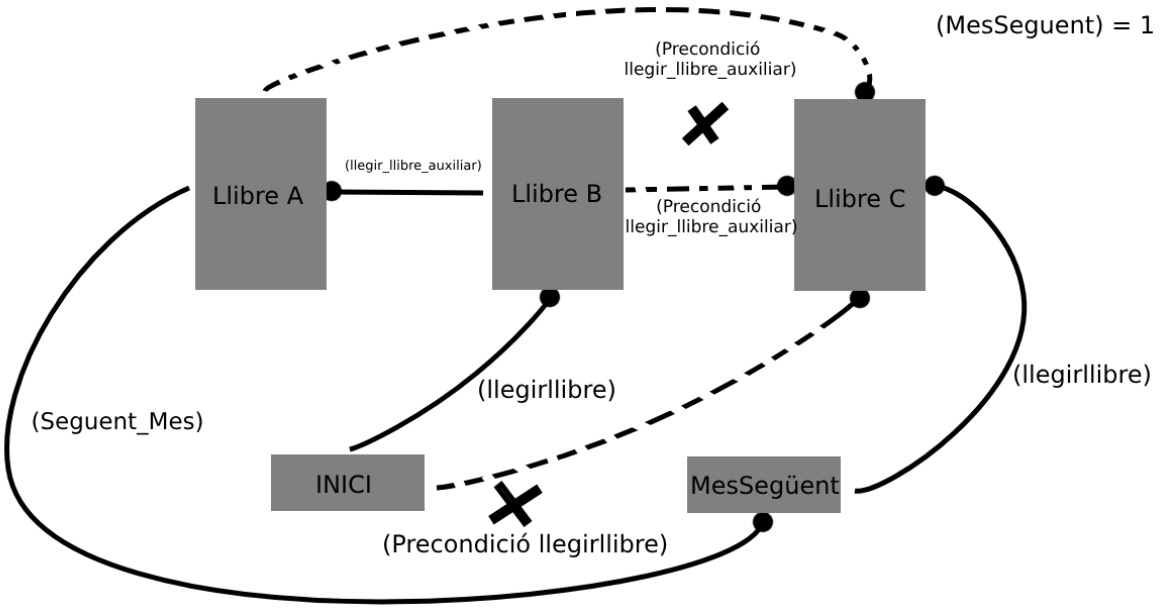
Això vol dir que, no podrà llegir el Llibre C, perquè té un predecessor que està marcat amb un d'aquests predicats, però sí que podrà llegir el Llibre A, que el marcarà com a llegit *(llegit Llibre A)* i també marcarà el llibre com a *(mes_anterior2 Llibre A)*.
Aquest segon predicat és necessari ja que si s'intenta fer només amb *(mes_anterior ?ll)*, l'acció *llegir_llibre_auxiliar* es retroalimentarà i llegirà tots els llibres alhora.



Una vegada ha llegit el Llibre B i el LLibre A al mateix més, ja no podrà utilitzar l'acció *llegir_llibre_auxiliar*, i només li queda l'opció de canviar de mes. Aquesta acció no te paràmetres, i els seus efectes son que, reinicia la variable *MesSegüent* a zero, i elimina tots els predicats existents de *(mes_anterior ?ll)* o *(mes_anterior2 ?ll)*.



Així, el cicle comença de zero un altre cop, i per últim llegirà el Llibre C, ja sigui amb *llegir_llibre* o *llegir_llibre_auxiliar*.



Ara anem a veure com son les accions, per tal de que funcionin així:

***Acció llegir_llibre*:**

Paràmetres: només un objecte llibre.

Precondició:
`(= (MesSeguent) 0)`
Aquesta condició fa que només s'executi una vegada per mes.

`(not (exists (?p) (and (predecessor ?p ?ll) (mes_anterior2 ?p))))`
Fa que no llegeixi llibres marcats com *(mes_anterior2 ?ll)* en cas de que s'executi l'acció *llegir_llibre_auxiliar* primer.

`(not (exists (?p) (and (predecessor ?p ?ll) (not (llegit ?p)))))`
Aquesta és la mateixa condició que ja hem vist abans al nivell bàsic.

`(delCataleg ?ll)`
Mira que el llibre siqui del catàleg.

Efectes:
`(llegit ?ll)`
Marca com llegit al llibre.

`(increase (MesSeguent) 1)`
Es podria dir que "puja la bandera" al planificador perquè canviï de mes quan vulgui.

`(mes_anterior ?ll)`
Marca el llibre com amb el predicat.

Acció llegir_llibre_auxiliar:

Paràmetres: només un llibre.

Precondició:
`(not (exists (?p) (and (predecessor ?p ?ll) (mes_anterior2 ?p))))`
Mira que no existeixi un predecessor del llibre que estigui marcat per *(mes_anterior2 ?ll)*.

`(not (exists (?p) (and (predecessor ?p ?ll) (mes_anterior ?p))))`
Mira que no existeixi un predecessor del llibre que estigui marcat per *(mes_anterior ?ll)*.

`(not (exists (?p) (and (predecessor ?p ?ll) (not (llegit ?p)))))`
Aquesta és la mateixa condició que ja hem vist abans al nivell bàsic.

`(delCataleg ?ll)`
Mira que el llibre siqui del catàleg

Efectes:
`(llegit ?ll)`
Marca com llegit al llibre.

`(mes_anterior ?p)`
Marca el llibre com amb el predicat.

Acció Seguent_Mes:

Paràmetres: cap.

Precondició:
`(= (MesSeguent) 1)`
Mira que la "bandera" estigui pujada.

`(exists (?ll) (or (mes_anterior ?ll) (mes_anterior2 ?ll)))`
Aquest predicat serveix perquè no canviï de mes quan no toca.

Efectes:

`(decrease (MesSeguent) 1)`
Torna a posar el nostre "booleà" a zero.

`(forall (?ll) (when (mes_anterior ?ll) (not (mes_anterior ?ll))))`
Aquí desfà tots el predicats de *(mes_anterior ?ll)*.

`(forall (?ll) (when (mes_anterior2 ?ll) (not (mes_anterior2 ?ll))))`
Aquí desfà tots el predicats de *(mes_anterior2 ?ll)*.

El domini ja està acabat, ara només faltarà crear el nostre arxiu problema:
Tenim tres sagues, Percy Jackson, Harry Potter i els apunts de GIA:

```
HP_I_La_Pedra_Filosofal        HP_I_La_Cambra_Secreta
HP_I_El_Pres_De_Azkaban        HP_I_El_Calze_De_foc
HP_I_La_Ordre_Del_Fenix        HP_I_El_Misteri_Del_Princep
HP_I_Les_Reliquies_De_La_Mort

PJ_El_Lladre_Del_Llampec       PJ_El_Mar_Dels_Monstres
PJ_La_Malediccio_Del_Tita            PJ_La_Batalla_Del_Laberint
PJ_L_Ultim_Heroi_De_L_Olimp          PJ_El_Calze_Dels_Deus

Apunts_PA1 Apunts_PA2 Apunts_CRA Apunts_FM Apunts_ABIA
```

Una vegada definits com a objectes, cal fer el init:
Primer s'han de definir tots el llibres que vulguem al catàleg en el nostre cas, dos de cada saga,

```
(delCataleg HP_I_La_Pedra_Filosofal)
(delCataleg HP_I_La_Cambra_Secreta)
(delCataleg PJ_El_Lladre_Del_Llampec)
(delCataleg PJ_El_Mar_Dels_Monstres)
 …
(delCatàleg Apunts_ABIA)
```

Després els predecessors, (els apunts d'ABIA tenen varis predecessors)

```
(predecessor HP_I_La_Pedra_Filosofal HP_I_La_Cambra_Secreta)
(predecessor PJ_El_Lladre_Del_Llampec PJ_El_Mar_Dels_Monstres)
 …
(predecessor Apunts_PA1 Apunts_PA2)
(predecessor Apunts_PA2 Apunts_ABIA)
(predecessor Apunts_CRA Apunts_ABIA)
(predecessor Apunts_FM Apunts_ABIA)
```

Init acabat, ara toca només fer el goal i el nostre problema ja estaria llest per funcionar:
Aquí podem posar que llegeixi dues sagues per comprovar que encara funciona pel domini anterior: (Arxiu *problema1_0.pddl*)

```
(llegit HP_I_Les_Reliquies_De_La_Mort)
(llegit PJ_El_Calze_Dels_Deus)
```

O les tres, per veure si serveix pel domini 1: (Arxiu *problema1_1.pddl*)

```
(llegit HP_I_Les_Reliquies_De_La_Mort)
(llegit PJ_El_Calze_Dels_Deus)
(llegit Apunts_ABIA)
```

Resultats del problema1_0: (metricff.exe -o domini1.pddl -f problema1_0.pddl)

```
0: LLEGIR_LLIBRE_AUXILIAR HP_I_LA_PEDRA_FILOSOFAL
```

```
1: LLEGIR_LLIBRE PJ_EL_LLADRE_DEL_LLAMPEC
2: SEGUENT_MES
3: LLEGIR_LLIBRE_AUXILIAR HP_I_LA_CAMBRA_SECRETA
4: LLEGIR_LLIBRE PJ_EL_MAR_DELS_MONSTRES
5: SEGUENT_MES
6: LLEGIR_LLIBRE_AUXILIAR HP_I_EL_PRES_DE_AZKABAN
7: LLEGIR_LLIBRE PJ_LA_MALEDICCIO_DEL_TITA
8: SEGUENT_MES
9: LLEGIR_LLIBRE_AUXILIAR HP_I_EL_CALZE_DE_FOC
10: LLEGIR_LLIBRE PJ_LA_BATALLA_DEL_LABERINT
11: SEGUENT_MES
12: LLEGIR_LLIBRE_AUXILIAR HP_I_LA_ORDRE_DEL_FENIX
13: LLEGIR_LLIBRE PJ_L_ULTIM_HEROI_DE_L_OLIMP
14: SEGUENT_MES
15: LLEGIR_LLIBRE_AUXILIAR PJ_EL_CALZE_DELS_DEUS
16: LLEGIR_LLIBRE HP_I_EL_MISTERI_DEL_PRINCEP
17: SEGUENT_MES
18: LLEGIR_LLIBRE_AUXILIAR HP_I_LES_RELIQUIES_DE_LA_MORT
```

Veiem que la nova acció de *llegir_llibre_auxiliar* es complementa perfectament amb l'acció de *llegir_llibre*, a més, també es pot apreciar que l'acció de canviar de més es realitza també, correctament.

Resultats del problema1_1: (metricff.exe -o domini1.pddl -f problema1_1.pddl)

```
0: LLEGIR_LLIBRE_AUXILIAR HP_I_LA_PEDRA_FILOSOFAL
1: LLEGIR_LLIBRE_AUXILIAR PJ_EL_LLADRE_DEL_LLAMPEC
2: LLEGIR_LLIBRE_AUXILIAR APUNTS_PA1
3: LLEGIR_LLIBRE_AUXILIAR APUNTS_CRA
4: LLEGIR_LLIBRE APUNTS_FM
5: SEGUENT_MES
6: LLEGIR_LLIBRE_AUXILIAR HP_I_LA_CAMBRA_SECRETA
7: LLEGIR_LLIBRE PJ_EL_MAR_DELS_MONSTRES
8: SEGUENT_MES
9: LLEGIR_LLIBRE_AUXILIAR HP_I_EL_PRES_DE_AZKABAN
10: LLEGIR_LLIBRE PJ_LA_MALEDICCIO_DEL_TITA
11: SEGUENT_MES
12: LLEGIR_LLIBRE_AUXILIAR HP_I_EL_CALZE_DE_FOC
13: LLEGIR_LLIBRE PJ_LA_BATALLA_DEL_LABERINT
14: SEGUENT_MES
15: LLEGIR_LLIBRE_AUXILIAR HP_I_LA_ORDRE_DEL_FENIX
16: LLEGIR_LLIBRE APUNTS_PA2
17: SEGUENT_MES
18: LLEGIR_LLIBRE_AUXILIAR APUNTS_ABIA
19: LLEGIR_LLIBRE_AUXILIAR PJ_L_ULTIM_HEROI_DE_L_OLIMP
20: LLEGIR_LLIBRE HP_I_EL_MISTERI_DEL_PRINCEP
21: SEGUENT_MES
22: LLEGIR_LLIBRE_AUXILIAR PJ_EL_CALZE_DELS_DEUS
23: LLEGIR_LLIBRE_AUXILIAR HP_I_LES_RELIQUIES_DE_LA_MORT
```

Aquí volíem comprovar, que, si un llibre té múltiples predecessors, tot funcionés com era esperat. Doncs així ha sigut, ja que les accions 2, 3 i 4 veiem que s'efectuen correctament i en l'ordre que esperàvem..

Tercer Domini (Extensió 2):

Ara quins són els requisits?

Extensió 2: *Extensió 1+ els llibres poden tenir de 0 a M llibres paral·lels. El planner és capaç de construir un pla per poder arribar a llegir els llibres objectiu, on per a tot llibre que pertany al pla, tots els seus llibres paral·lels pertanyen al pla i són al mateix mes o en mesos anteriors.*

Ara només hem d'implementar els llibres paral·lels. Com ho farem?

Pues respecte al domini anterior, no hi ha gaire canvi:

<u>En quant a requisits:</u>
(el requisit *stripes* és un basic indispensable)

Si pensem en posar el requisit *typing*, passa el mateix que en l'anterior nivel, i per tant no ho farem.

Ara anem amb el requisit *fluents*. L'utilitzarem pel següent motiu, utilitzarem un fluent com a variable booleana, per així poder saber quan canviar de més i poder guiar al nostre planificador. (Com ja hem explicat al domini anterior)

Com utilitzarem de base al planner anterior, necessitarem obligatòriament el requisit *adl*.

<u>Ara amb els predicats, només un nou:</u>

(parallel ?ll1 ?ll2)

El predicat funciona de la següent manera:
Segons la nostra implementació, el llibre *?ll1* serà paral·lel a *?ll2*.

Doncs ja estem amb els predicats. Com hem vist fins ara, l'únic canvi que hi ha hagut és la introducció d'un nou predicat, així que passem directament al funcionament del planificador en aquest domini:

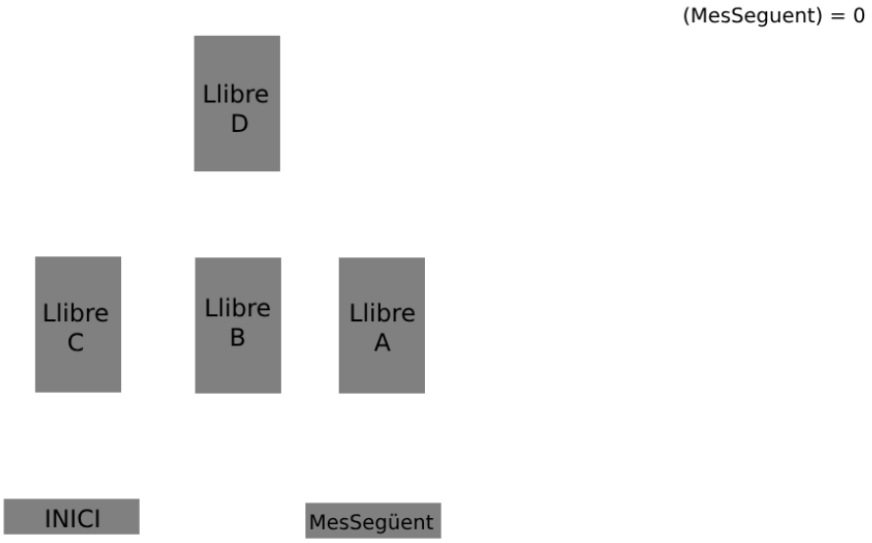
Les accions? les tres mateixes que abans, *llegir_llibre*, *llegir_llibre_auxiliar* i *Seguent_Mes*. Aleshores, què ha canviat? Doncs l'únic que ha canviat ha sigut un canvi a la precondició de *llegir_llibre*, només una línia nova, l'analitzarem després.

```
(forall (?para) (not (parallel ?para ?ll)))
```

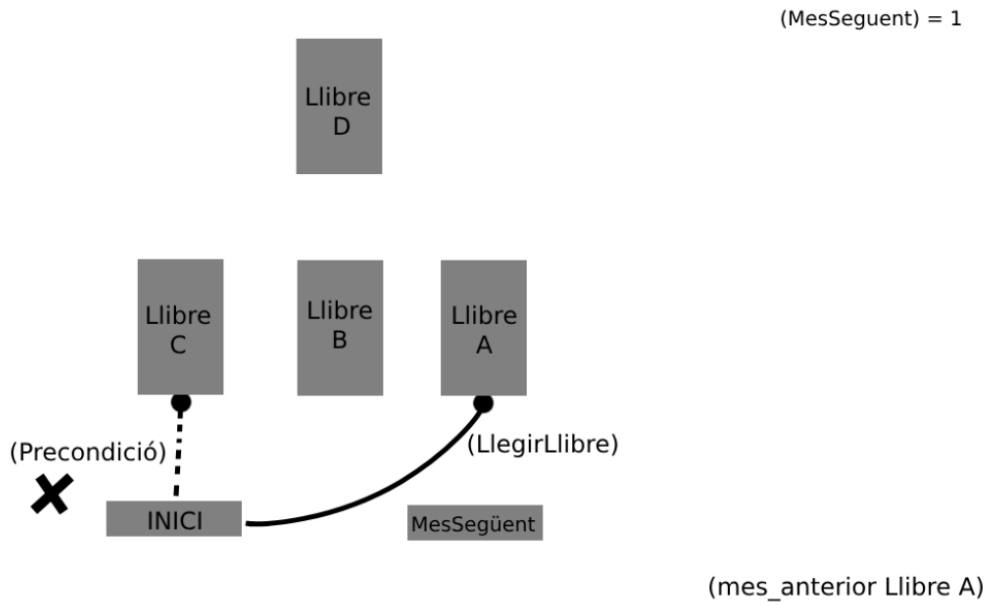
Anem a veure com funciona el planificador des de dins:

Principalment, funciona com el domini anterior però ara entren en joc els llibres paral·lels, i no hi ha una millor manera de veure-ho que amb un exemple:

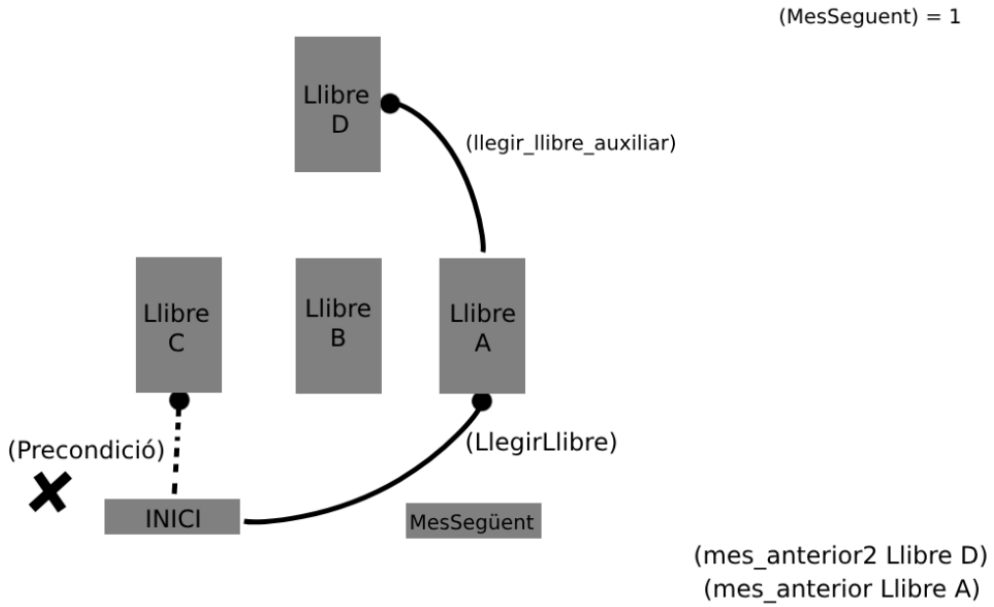
Tenim quatre llibres, el llibre A, el llibre B, el llibre C i el llibre D, sent el llibre D paral·lel al llibre C i el llibre A i el B predecessors del llibre C: (L'objectiu és llegir el llibre C)



Una vegada comença el planificador, llegeix idealment, amb l'acció *llegirllibre* el llibre A o el llibre B, nosaltres farem que sigui el llibre A:



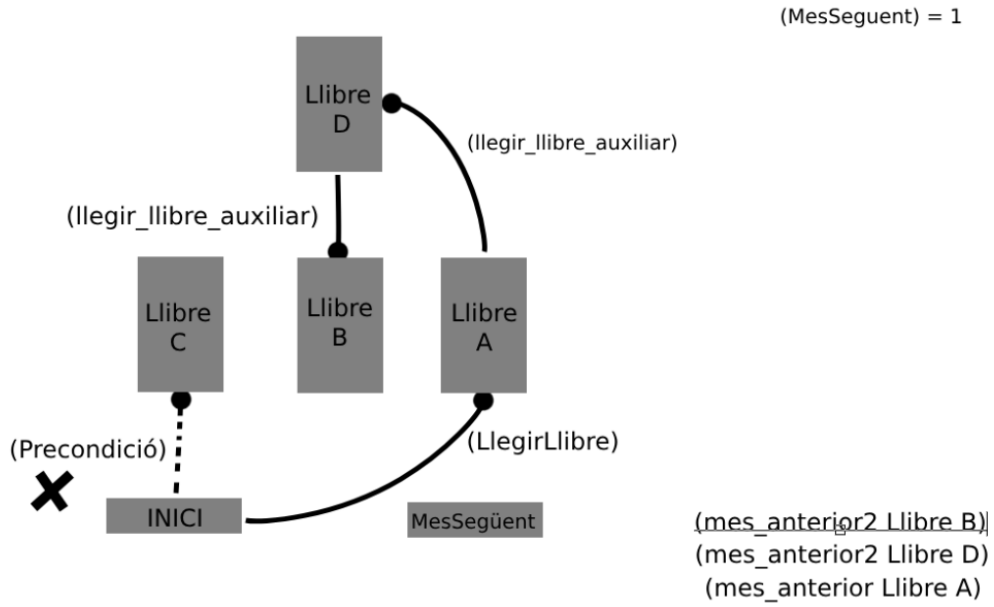
Ara que *llegirllibre* ja s'ha usat una vegada, el *flag* de *MesSeguent* ja està pujat i de moment, tot igual que a l'anterior domini, anem als següents passos:



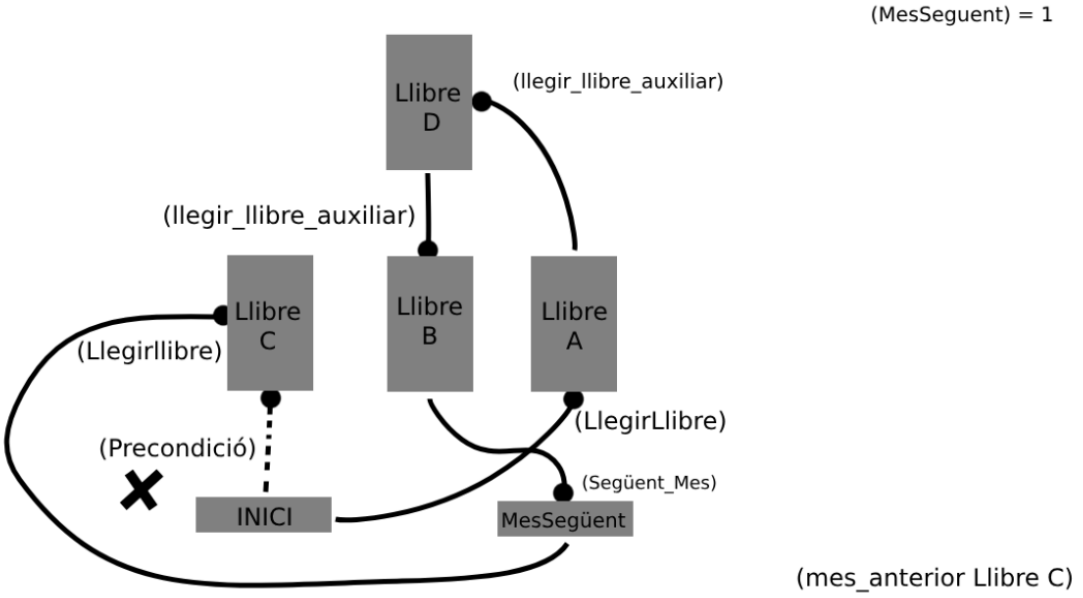
Ara ha llegit el llibre paral·lel, per què? Doncs perquè per les condicions que hi ha a sobre de la condició (`(forall (?para) (not (parallel ?para ?ll)))`) fan que, no es llegeixi cap llibre paral·lel en el mateix moment, quan diem al mateix moment, ens referim al fet que no els llegeix el mateix mes ni després, sempre abans.

Això vol dir que haver llegit el llibre D podia haver passat després de llegir el llibre B, què és el següent, però el llegirà després.

Així és com quedaria abans de canviar de mes:



Ara què toca? Doncs canviar de més per tal de que es pugui llegir el llibre C i s'acabi de planificar el nostre problema:



Anem a repassar l'acció de *llegirllibre*:

***Acció llegir_llibre*:**

Paràmetres: només un objecte llibre.
Precondició:
`(= (MesSeguent) 0)`
Aquesta condició fa que només s'executi una vegada per mes.

```
(not (exists (?p) (and (predecessor ?p ?ll) (mes_anterior2 ?p))))
```

Fa que no llegeixi llibres marcats com *(mes_anterior2 ?ll)* en cas de que s'executi l'acció *llegir_llibre_auxiliar* primer.

```
(not (exists (?p) (and (predecessor ?p ?ll) (not (llegit ?p)))))
```
Aquesta és la mateixa condició que ja hem vist abans al nivell bàsic.

```
(forall (?para) (not (parallel ?para ?ll)))
```
En combinació amb les anteriors, fa que no es llegeixi ningún llibre paral·lel en mateix mes.

```
(delCataleg ?ll)
```
Mira que el llibre siqui del catàleg.

Efectes:
```
(llegit ?ll)
```
Marca com llegit al llibre.

```
(increase (MesSeguent) 1)
```
Es podria dir que "puja la bandera" al planificador perquè canviï de mes quan vulgui.

```
(mes_anterior ?ll)
```
Marca el llibre com amb el predicat.

I com hem fet el nostre arxiu problema? Doncs no és gaire diferent al del domini anterior, només afegim el predicat *(parallel ?ll1 ?ll2)* i les seves posteriors relacions:

```
(parallel Els_Deus_Grecs PJ_El_Lladre_Del_Llampec)
(parallel Els_Deus_Grecs PJ_El_Mar_Dels_Monstres)
(parallel Els_Deus_Grecs PJ_La_Malediccio_Del_Tita)
(parallel Els_Deus_Grecs PJ_La_Batalla_Del_Laberint)
 …
(parallel KANE_El_Tron_De_Foc HO_La_Sang_De_L_Olimp)
(parallel KANE_L_Ombra_De_La_Serp HO_L_Heroi_Perdut)
(parallel KANE_L_Ombra_De_La_Serp HO_El_Fill_De_Neptu)
(parallel KANE_L_Ombra_De_La_Serp HO_La_Marca_D_Atena)
(parallel KANE_L_Ombra_De_La_Serp HO_La_Casa_D_Hades)
(parallel KANE_L_Ombra_De_La_Serp HO_La_Sang_De_L_Olimp)
```

Hem creat dos arxius, un per provar que l'anterior modificació permeti que el llibres sense paral·lels funcionin bé (arxiu *problema2_0.pddl*), és a dir, que no haguem espatllat el problema original:

```
(:goal
    (and
      (llegit HP_I_Les_Reliquies_De_La_Mort)
      (llegit Apunts_ABIA)
       …
```

I un altre per mirar si aquesta modificació és efectiva i fa el que ha de fer (arxiu *problema2_1.pddl*).

```
(:goal
    (and
      (llegit PJ_El_Calze_Dels_Deus)
```

```
(llegit Apunts_ABIA)
 …
```

Resultats del problema2_0: (metricff.exe -O -o domini2.pddl -f problema2_0.pddl)

```
0: LLEGIR_LLIBRE HP_I_LA_PEDRA_FILOSOFAL
1: LLEGIR_LLIBRE_AUXILIAR APUNTS_PA1
2: LLEGIR_LLIBRE_AUXILIAR APUNTS_CRA
3: LLEGIR_LLIBRE_AUXILIAR APUNTS_FM
4: SEGUENT_MES
5: LLEGIR_LLIBRE HP_I_LA_CAMBRA_SECRETA
6: LLEGIR_LLIBRE_AUXILIAR APUNTS_PA2
7: SEGUENT_MES
8: LLEGIR_LLIBRE_AUXILIAR APUNTS_ABIA
9: LLEGIR_LLIBRE HP_I_EL_PRES_D_AZKABAN
10: SEGUENT_MES
11: LLEGIR_LLIBRE HP_I_EL_CALZE_DE_FOC
12: SEGUENT_MES
13: LLEGIR_LLIBRE HP_I_LA_ORDRE_DEL_FENIX
14: SEGUENT_MES
15: LLEGIR_LLIBRE HP_I_EL_MISTERI_DEL_PRINCEP
16: SEGUENT_MES
17: LLEGIR_LLIBRE HP_I_LES_RELIQUIES_DE_LA_MORT
```

Volíem mirar si quan no hi havia llibres paral·lels tot continuava funcionant com abans, i així ha sigut.

Resultats del problema2_1: (metricff.exe -O -o domini2.pddl -f problema2_1.pddl)

```
0: LLEGIR_LLIBRE APUNTS_PA1
1: LLEGIR_LLIBRE_AUXILIAR PJ_EL_LLADRE_DEL_LLAMPEC
2: LLEGIR_LLIBRE_AUXILIAR APUNTS_CRA
3: LLEGIR_LLIBRE_AUXILIAR APUNTS_FM
4: SEGUENT_MES
5: LLEGIR_LLIBRE APUNTS_PA2
6: SEGUENT_MES
7: LLEGIR_LLIBRE APUNTS_ABIA
8: LLEGIR_LLIBRE_AUXILIAR PJ_EL_MAR_DELS_MONSTRES
9: SEGUENT_MES
10: LLEGIR_LLIBRE KANE_LA_PIRAMIDE_VERMELLA
11: LLEGIR_LLIBRE_AUXILIAR PJ_LA_MALEDICCIO_DEL_TITA
12: SEGUENT_MES
13: LLEGIR_LLIBRE KANE_EL_TRON_DE_FOC
14: LLEGIR_LLIBRE_AUXILIAR PJ_LA_BATALLA_DEL_LABERINT
15: SEGUENT_MES
16: LLEGIR_LLIBRE KANE_L_OMBRA_DE_LA_SERP
17: LLEGIR_LLIBRE_AUXILIAR PJ_L_ULTIM_HEROI_DE_L_OLIMP
18: SEGUENT_MES
19: LLEGIR_LLIBRE ELS_HEROIS_GRECS
20: LLEGIR_LLIBRE_AUXILIAR HO_L_HEROI_PERDUT
21: SEGUENT_MES
22: LLEGIR_LLIBRE ELS_DEUS_GRECS
23: LLEGIR_LLIBRE_AUXILIAR HO_EL_FILL_DE_NEPTU
```

```
24: SEGUENT_MES
25: LLEGIR_LLIBRE HP_I_LA_PEDRA_FILOSOFAL
26: LLEGIR_LLIBRE_AUXILIAR HO_LA_MARCA_D_ATENA
27: SEGUENT_MES
28: LLEGIR_LLIBRE HP_I_LA_CAMBRA_SECRETA
29: LLEGIR_LLIBRE_AUXILIAR HO_LA_CASA_D_HADES
30: SEGUENT_MES
31: LLEGIR_LLIBRE HP_I_EL_PRES_D_AZKABAN
32: LLEGIR_LLIBRE_AUXILIAR HO_LA_SANG_DE_L_OLIMP
33: SEGUENT_MES
34: LLEGIR_LLIBRE PJ_EL_CALZE_DELS_DEUS
```

Quedava per mirar que si, hi han llibre paral·lels, tot funcionés bé. Nosaltres hem comprovat l'ordre de predecessors i paral·lels de *Percy Jackson* que hi ha a la **pàgina 3**, i ho fa correctament.

Encara així, honestament hem de dir que, pel funcionament del nostre domini, pot ser que llegeixi llibres que no haurien d'estar en el pla, però tampoc hi ha cap restricció a l'enunciat sobre això. Per això, com que no és gaire exagerat perquè de les tres sagues no les llegeix totes i és un error que no sol passar, si fos un error més greu, ho haguéssim arreglat.

Quart Domini (Extensió 3):

Extensió 3: *Els llibres tenen a més a més un nombre de pàgines. El planificador controla que al pla generat no se superin les 800 pàgines al mes.*

Hem d'afegir una manera de controlar el nombre de pàgines que té cada llibre i les pàgines que es llegeixen en un mes. Ho fem amb dos fluents: (PaginesLlibre ?ll) i (PaginesMes). PaginesLlibre és un fluent que està associat a cada llibre i s'inicialitza a l'init del problema així:

```
(= (PaginesLlibre PJ_El_Calze_Dels_Deus) 256)
```

PaginesMes no té cap paràmetre i s'actualitza a les accions llegir_llibre i llegir_llibre_auxiliar, on té com a precondició que el seu valor més PaginesLlibre del llibre no superi les 800 pàgines.

```
(< (+ (PaginesMes) (PaginesLlibre ?ll)) 801)
```

I a l'efecte de les dues condicions se sumen les pàgines del llibre llegit a PaginesMes.

```
(increase (PaginesMes) (PaginesLlibre ?ll))
```

Quan canviem de mes amb SeguentMes cal restablir PaginesMes a 0, ho fem restant el seu valor de si mateix.

```
(decrease (PaginesMes) (PaginesMes))
```

Ara que ja hem acabat el domini, hem de canviar dues coses al problema.

Primer de tot cal declarar el fluent PaginesLlibre i assignar-li un valor per a cada llibre del catàleg. Per exemple: `(= (PaginesLlibre PJ_El_Calze_Dels_Deus) 256)`. Per als fitxers problema, hem fet servir el nombre de pàgines reals dels llibres. També cal inicialitzar el fluent PaginesMes, que comença amb valor 0 per defecte, `(= (PaginesMes) 0)`.

Hem creat dos goals diferents per a dos fitxers (*problema3_0* i *problema3_1)*.

Goal de *problema3_0*: el goal és el mateix que el del problema2_1. Haurem de comprovar que les pàgines dels llibres que llegeix en un mateix mes no en superi les 800.

```
(and (llegit PJ_El_Calze_Dels_Deus) (llegit Apunts_ABIA))
```

Goal de *problema3_1:* només caldrà que llegeixi els llibres sense successors, ja que si sí que en té, el successor l'haurà de llegir per força (precondició de les accions de lectura). Això estalvia haver de fer tantes comparacions, i es nota molt amb problemes amb cadenes llargues.

```
(forall (?ll) (imply (not (exists (?ll2) (predecessor ?ll ?ll2))) (llegit ?ll)))
```

Resultats *problema3_0*: (metricff -O -o domini3.pddl -f problema3_0.pddl)

```
0: LLEGIR_LLIBRE APUNTS_PA1
1: LLEGIR_LLIBRE_AUXILIAR PJ_EL_LLADRE_DEL_LLAMPEC
2: SEGUENT_MES
3: LLEGIR_LLIBRE HP_I_LA_PEDRA_FILOSOFAL
4: LLEGIR_LLIBRE_AUXILIAR PJ_EL_MAR_DELS_MONSTRES
```

```
5: SEGUENT_MES
6: LLEGIR_LLIBRE KANE_LA_PIRAMIDE_VERMELLA
7: SEGUENT_MES
8: LLEGIR_LLIBRE KANE_EL_TRON_DE_FOC
9: LLEGIR_LLIBRE_AUXILIAR PJ_LA_MALEDICCIO_DEL_TITA
10: SEGUENT_MES
11: LLEGIR_LLIBRE ELS_DEUS_GRECS
12: LLEGIR_LLIBRE_AUXILIAR PJ_LA_BATALLA_DEL_LABERINT
13: SEGUENT_MES
14: LLEGIR_LLIBRE KANE_L_OMBRA_DE_LA_SERP
15: LLEGIR_LLIBRE_AUXILIAR PJ_L_ULTIM_HEROI_DE_L_OLIMP
16: SEGUENT_MES
17: LLEGIR_LLIBRE APUNTS_PA2
18: LLEGIR_LLIBRE_AUXILIAR HO_L_HEROI_PERDUT
19: SEGUENT_MES
20: LLEGIR_LLIBRE HP_I_LA_CAMBRA_SECRETA
21: LLEGIR_LLIBRE_AUXILIAR HO_EL_FILL_DE_NEPTU
22: SEGUENT_MES
23: LLEGIR_LLIBRE APUNTS_CRA
24: LLEGIR_LLIBRE_AUXILIAR HO_LA_MARCA_D_ATENA
25: SEGUENT_MES
26: LLEGIR_LLIBRE APUNTS_FM
27: LLEGIR_LLIBRE_AUXILIAR HO_LA_CASA_D_HADES
28: SEGUENT_MES
29: LLEGIR_LLIBRE APUNTS_ABIA
30: LLEGIR_LLIBRE_AUXILIAR HO_LA_SANG_DE_L_OLIMP
31: SEGUENT_MES
32: LLEGIR_LLIBRE PJ_EL_CALZE_DELS_DEUS
```

Si fem la suma de les pàgines que llegeix cada mes, veiem que no superen les 800 pàgines (499, 589, 504, 748, 800, 763, 718, 755, 645, 657, 686, 256), i a més a més, fa combinacions semiòptimes per aprofitar els mesos (llibres llargs i llibres curts el mateix mes). Igual que abans, llegeix algun llibre de Harry Potter que no caldria, però només passa 2 cops.

Resultats *problema3_1:* (metricff -O -o domini3.pddl -f problema3_1.pddl)

```
0: LLEGIR_LLIBRE HP_I_LA_PEDRA_FILOSOFAL
1: LLEGIR_LLIBRE_AUXILIAR PJ_EL_LLADRE_DEL_LLAMPEC
2: LLEGIR_LLIBRE_AUXILIAR APUNTS_PA1
3: SEGUENT_MES
4: LLEGIR_LLIBRE ELS_HEROIS_GRECS
5: LLEGIR_LLIBRE_AUXILIAR PJ_EL_MAR_DELS_MONSTRES
6: SEGUENT_MES
7: LLEGIR_LLIBRE KANE_LA_PIRAMIDE_VERMELLA
8: SEGUENT_MES
9: LLEGIR_LLIBRE KANE_EL_TRON_DE_FOC
10: LLEGIR_LLIBRE_AUXILIAR PJ_LA_MALEDICCIO_DEL_TITA
11: SEGUENT_MES
12: LLEGIR_LLIBRE ELS_DEUS_GRECS
13: LLEGIR_LLIBRE_AUXILIAR PJ_LA_BATALLA_DEL_LABERINT
14: SEGUENT_MES
```

```
15: LLEGIR_LLIBRE KANE_L_OMBRA_DE_LA_SERP
16: LLEGIR_LLIBRE_AUXILIAR PJ_L_ULTIM_HEROI_DE_L_OLIMP
17: SEGUENT_MES
18: LLEGIR_LLIBRE APUNTS_PA2
19: LLEGIR_LLIBRE_AUXILIAR HO_L_HEROI_PERDUT
20: SEGUENT_MES
21: LLEGIR_LLIBRE HP_I_LA_CAMBRA_SECRETA
22: LLEGIR_LLIBRE_AUXILIAR HO_EL_FILL_DE_NEPTU
23: SEGUENT_MES
24: LLEGIR_LLIBRE HP_I_EL_PRES_D_AZKABAN
25: SEGUENT_MES
26: LLEGIR_LLIBRE HP_I_EL_CALZE_DE_FOC
27: SEGUENT_MES
28: LLEGIR_LLIBRE HP_I_LA_ORDRE_DEL_FENIX
29: SEGUENT_MES
30: LLEGIR_LLIBRE HP_I_EL_MISTERI_DEL_PRINCEP
31: SEGUENT_MES
32: LLEGIR_LLIBRE HP_I_LES_RELIQUIES_DE_LA_MORT
33: SEGUENT_MES
34: LLEGIR_LLIBRE APUNTS_CRA
35: LLEGIR_LLIBRE_AUXILIAR HO_LA_MARCA_D_ATENA
36: SEGUENT_MES
37: LLEGIR_LLIBRE APUNTS_FM
38: LLEGIR_LLIBRE_AUXILIAR HO_LA_CASA_D_HADES
39: SEGUENT_MES
40: LLEGIR_LLIBRE APUNTS_ABIA
41: LLEGIR_LLIBRE_AUXILIAR HO_LA_SANG_DE_L_OLIMP
42: SEGUENT_MES
43: LLEGIR_LLIBRE_AUXILIAR MAGS_I_SEMIDEUS
44: LLEGIR_LLIBRE PJ_EL_CALZE_DELS_DEUS
```

Aquest és el problema “definitiu”. El pla generat inclou tots els apunts d’assignatures i els llibres de la saga de HP i l’univers de PJ. L’ordre dels llibres és correcte, no hi ha cap violació ni dels predecessors ni dels paral·lels, i tampoc es passa de 800 pàgines per mes. El pla acaba llegint Apunts_ABIA, Mags i Semideus i PJ i El Calze dels Déus en els últims dos mesos, que són les cues de les cadenes de llibres de PJ i Apunts. El pla acaba HP uns quants mesos abans.

Generador de problemes

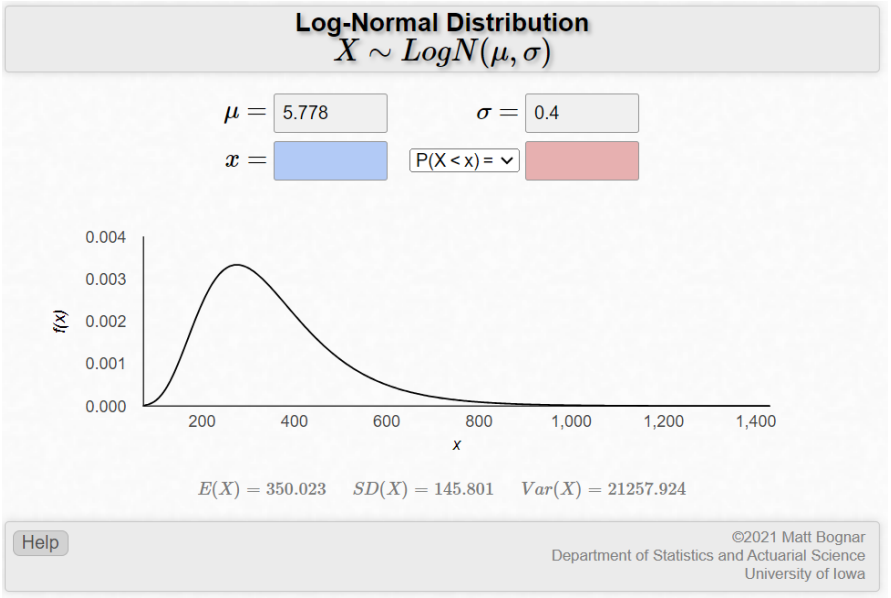
Els problemes es generen cridant una funció `generar_problema(extensió=3, n=10, llavor=42)`. La mida del problema ha de ser entre 2 i 260 llibres. Els llibres es guarden en una llista de mida n i es diuen *[A0, A1, A2, …, A9, B1, B2, …, Z9].* La llavor s'utilitza per a poder reproduir els resultats dels nombres aleatoris. Després es crea un fitxer problema p{extensió}_{n}.pddl i es defineix el domini, el problema i els objectes, és a dir els llibres.

A l'init s'inclouen els predicats *delCataleg* per a tots els dominis els fluents i *PaginesLlibre* per a l'extensió 3.

El nombre de pàgines generades aleatòriament segueix una distribució log-normal, amb una mitjana de 350 pàgines i una desviació estàndard de 0,4. Els valors els hem triat després de provar-ne uns quants a aquesta pàgina web i mirar que tinguessin sentit. https://homepage.divms.uiowa.edu/~mbognar/applets/lognormal.html

$$E(X) = e^{\mu+\sigma^2/2} \Rightarrow ln(E(X)) = \mu + \sigma^2/2 \Rightarrow \mu = ln(E(X)) - \sigma^2/2 = ln(350) - 0.4^2/2$$

$$\mu \approx 5,778$$



La mediana esperada és de $e^{5,778} \approx 323$, que és raonable i hi ha una probabilitat que es generi un número més gran que 800 o més petit que 50 d'1,17% i pràcticament 0%, que simplement els convertim a 800 i 50 per simplificar. Per resumir, el nombre de pàgines segueixen una fórmula així:

```
x = min(max(50, int(random.lognormvariate(5.733, 0.5))), 800)
```

Els predecessors, que n'hi ha un nombre aleatori (com a màxim n). Els llibres es tracten com si ja tinguessin un ordre preestablert: *A0→A1→…* i així no cal comprovar si hi ha cicles, perquè sempre serà un arbre o conjunt d'arbres, un llibre posterior no podrà ser predecessor d'un llibre anterior. Per al nivell bàsic també hem de comprovar que no es repeteixin els llibres successors (hi ha un límit d'un). Els llibres paral·lels són similars, però els limitem a l'arrel quadrada de la

mida del problema perquè és més realista, normalment no hi ha gaires sèries que tinguin llibres paral·lels. També establim aquest ordre, o sigui que no hi haurà un paral·lel *B0* paral·lel d'*A0*, i no es repetiran (no passaria res, però igualment ho comprovem). Per acabar, inicialitzem MesSeguent i PaginesMes a 0 si correspon.

El goal serà el mateix per a tots:

```
(forall (?ll) (imply (not (exists (?ll2) (predecessor ?ll ?ll2))) (llegit ?ll)))
```

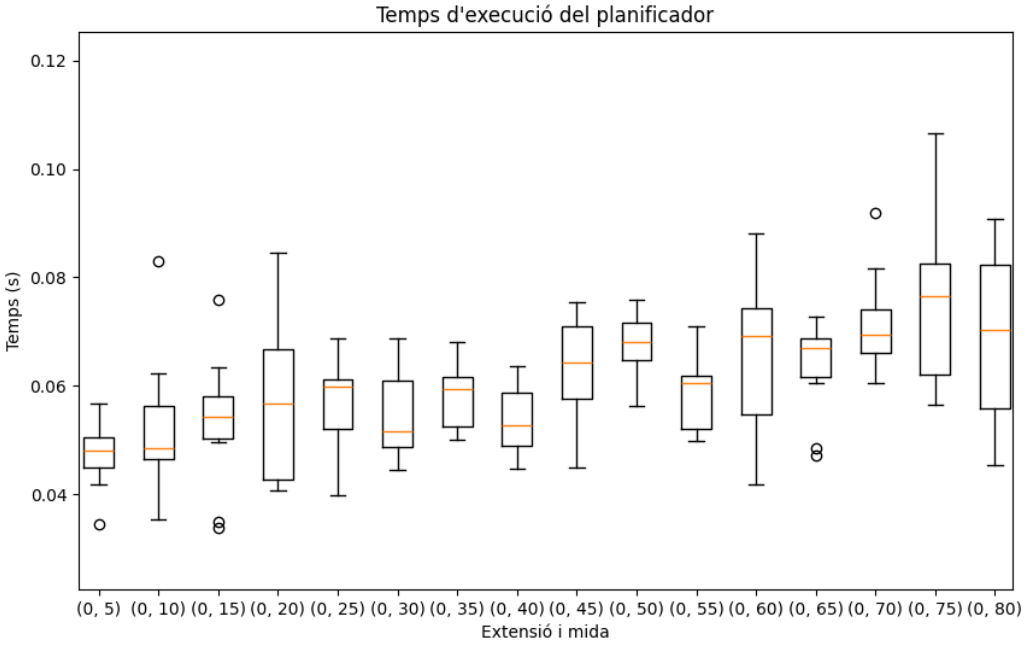
El pla acabarà quan hagi llegit tots els llibres sense successors, però perquè això passi caldrà que llegeixi tots els llibres del catàleg. La mètrica d'optimització serà el plan-length, que ja va prou bé: evita que es passin els mesos innecessàriament, es llegeix tant en un mes com es pot i en general no li agrada que es llegeixin llibres si no són estrictament necessaris.

Experiments

Els experiments els hem realitzat amb 10 execucions (llavors 0-9) per a cada extensió (0-3) i mida (5-80, increment de 5). Hem fet servir el temps calculat amb la biblioteca time, ja que el comptador de temps de metricff no és gaire fiable a Windows. Hem limitat el temps màxim, però només ha calgut per a les execucions de problemes més grans (> 55 llibres, i només uns quants).

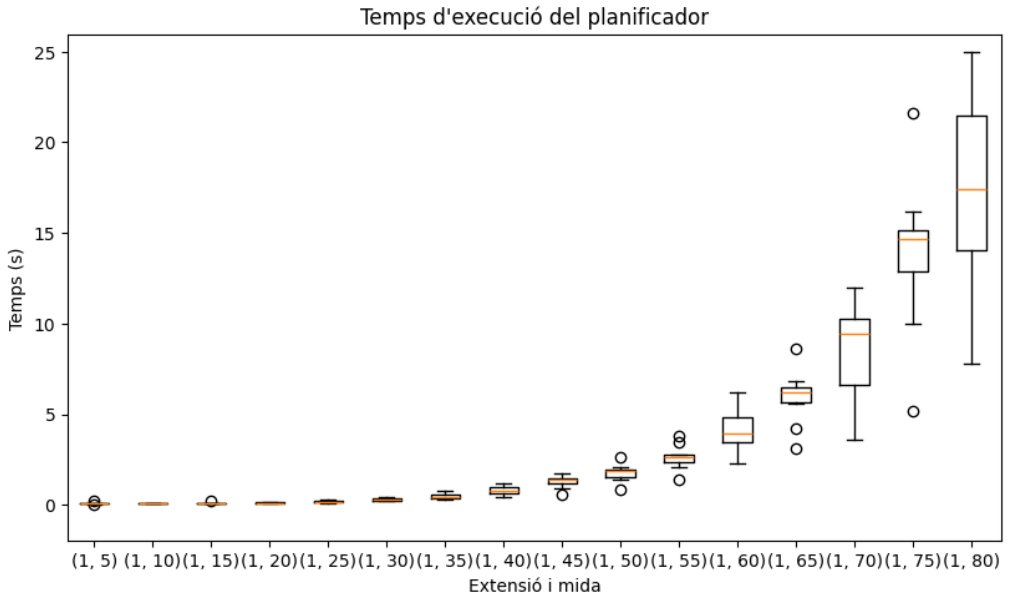
Després hem visualitzat el creixement del temps amb histogrames per mirar com creix la variància.

Extensió 0



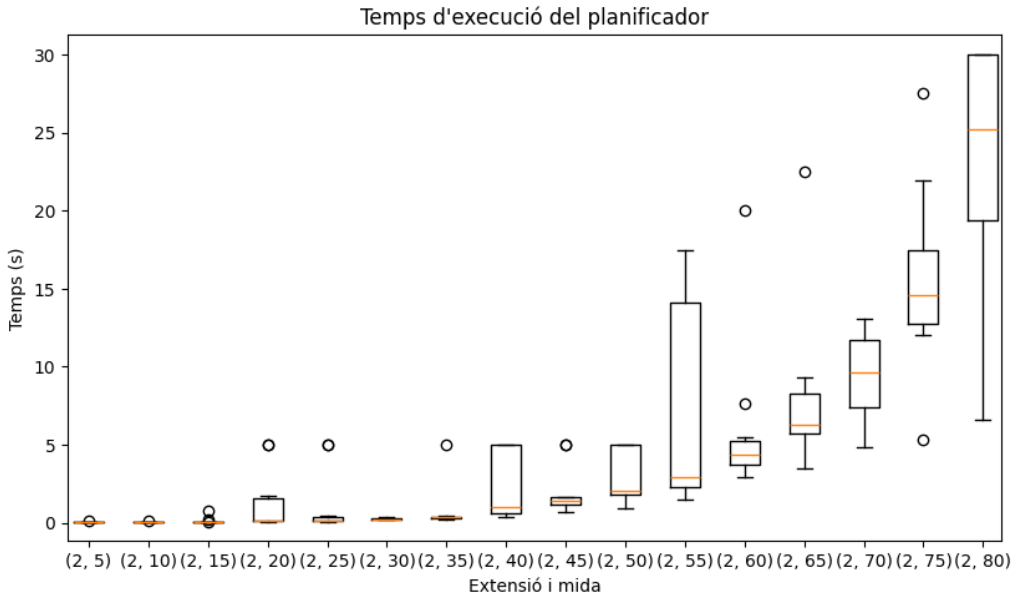
L'extensió 0 triga molt poc en totes les mides que hem provat. Es pot observar un patró lineal moderat creixent però hi ha molta variació entre les execucions d'una mateixa mida.

Extensió 1



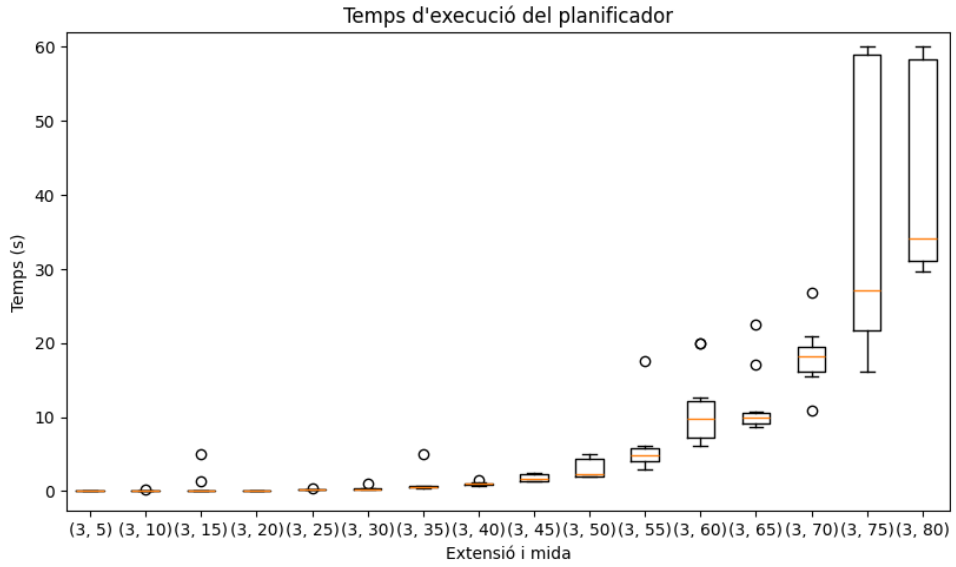
A l'extensió 1 ja podem observar el creixement exponencial que esperaríem d'aquesta mena de problemes. La mida de les capses també creix molt significativament.

Extensió 2



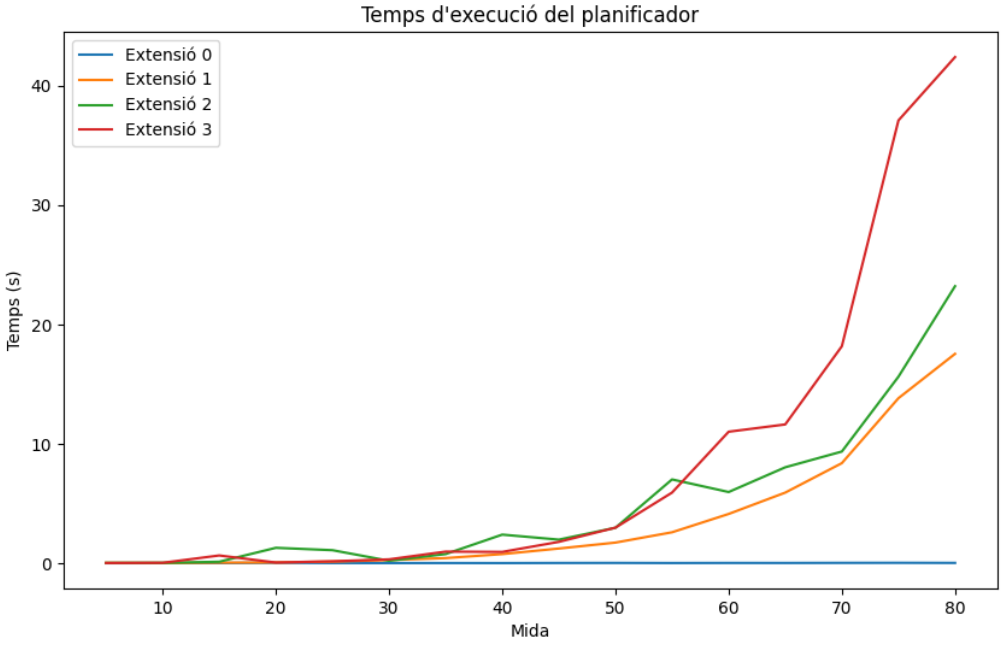
Seguim veient el comportament exponencial però és més pronunciat i comença a pujar abans. També hi podem veure més outliers.

Extensió 3



Mateix comportament exponencial, però aquí podem veure el límit de temps imposat (60s) per execució, que limita als problemes amb 75 i 80 llibres.

També hem fet un diagrama on podem veure totes les extensions alhora en una mateixa escala.



Podem inferir que la complexitat temporal del nivell bàsic és $O(m)$ i la de les extensions és $O(k^m)$, on k és una constant (diferent per a cada extensió) i m és la mida/número de llibres.

Conclusions

Els problemes amb què hem treballat han estat una petita part del camp de Planificació Automàtica. Hem fet Planificació Clàssica. Concretament, hem treballat amb problemes deterministes, estat inicial únic, d'informació perfecta i accions asíncrones amb un sol agent.

Podem concloure que els planificadors dels diferents dominis ens permeten generar plans vàlids per a sèries de llibres amb un ordre simple o complex. Una limitació imposada per l'enunciat és que no podem representar sèries on hi ha cicles (principi és el final), però podria ser un camí per continuar generalitzant el planificador.

Els plans generats pels dominis són tots satisfactibles (o falsos), és a dir, que no garanteixen que siguin els òptims. La versió optimitzada és un algorisme Best-first Search, per tant, encara que h sigui admissible, no ens garanteix que sigui un pla òptim.

La cerca de plans té una complexitat temporal exponencial per a les extensions. Això passa perquè a cada node les accions han de fer comprovacions per a tots els altres llibres del catàleg, mentre que al nivell bàsic només ho fan per a 1 predecessor com a màxim. És a dir, que l'espai d'estats de les extensions "explota" i creix exponencialment, mentre que el nivell bàsic està limitat pel nombre de predecessors $(1^k) = 1 \quad \forall k$.