

Pràctica Planificació PDDL

Algorismes Bàsics per la Intel·ligència Artificial



Joan Bennàssar Martín
Joan Bernaus Casadesús
Jaume Mora i Ladària

GRUP 11 - ABIA
Grau en Intel·ligència Artificial
Universitat Politècnica de Catalunya
Barcelona, a desembre de 2023

Índex

| | |
|---|-----------|
| Introducció..... | 2 |
| Desenvolupament dels models..... | 3 |
| Modelat del domini..... | 4 |
| Extensió bàsica:..... | 4 |
| Extensió 1:..... | 6 |
| Extensió 2:..... | 8 |
| Extensió 3:..... | 9 |
| Modelat dels problemes..... | 10 |
| Conjunt de problemes de prova i resultats..... | 11 |
| Extensió 3 problema 2:..... | 17 |
| Generador de problemes..... | 18 |
| Generador per l'extensió 2..... | 18 |
| Generador per l'extensió 3..... | 19 |
| Utilització..... | 19 |
| Experimentació..... | 20 |
| Experiment 1..... | 20 |
| Experiment 2..... | 22 |
| Experiment 3..... | 24 |
| Experiment 4..... | 26 |
| Experiment 5..... | 28 |
| Conclusions i satisfacció general..... | 30 |

Introducció

En aquest document podem trobar la documentació referent a les descripcions i justificacions de la implementació que hem realitzat així com dels resultats que hem obtingut a la pràctica de PDDL realitzada per en Joan Bennàssar, en Joan Bernaus i en Jaume Mora per a l'assignatura d'Algoritmes Bàsics de la Intel·ligència Artificial.

L'objectiu principal d'aquesta pràctica de laboratori ha estat familiaritzar-se amb la resolució de problemes mitjançant un llenguatge nou per nosaltres com és el PDDL.

Al problema a resoldre se'ns demana que fem una planificació dels llibres a llegir per una persona durant un any. S'hi afegeix a la complicació del problema que alguns llibres són predecessors d'uns altres (per tant, s'han d'haver llegit anteriorment) i d'altres són paral·lels entre si (per tant, es poden llegir amb màxim un mes de diferència).

Com que hem decidit implementar totes les extensions, finalment hem afegit la complicació de les pàgines la qual ens limita que no es puguin llegir més de 800 pàgines al mes.

A l'haver fet totes les extensions amb tots els jocs de prova corresponents demanats trobareu dos fitxers de jocs de proves per cada extensió (la bàsica l'hem anomenat extensió 0). A més a més, com explicarem posteriorment hem creat dos dominis diferents per cada una de les extensions 0 i 1 i un domini per les extensions 2 i 3.

Desenvolupament dels models

Per a desenvolupar els diferents models hem seguit un procés iteratiu. Inicialment, durant la primera setmana, ens vam abocar a les extensions 0 i 1, i posteriorment, a la segona setmana, ens vam centrar en les extensions 2 i 3.

Vam començar el nostre treball creant un domini que assignava posicions als diferents mesos (amb gener a la posició 0, febrer a la posició 1 i així successivament). Malauradament, vam tenir dificultats a l'extensió 2 que ens van impedir gestionar adequadament els mesos assignats en posicions i els llibres paral·lels alhora.

Amb la finalitat d'evitar estancar-nos, com ens havia passat en la pràctica anterior vam decidir començar el domini de nou. Així, vam optar per un enfocament que entrellacés els mesos entre si (mes_següent gener febrer, mes_següent febrer març, etc.).

Al observar que aquest nou enfocament ens permetia resoldre de manera eficient el problema dels llibres paral·lels, vam adaptar el domini per les extensions 0 i 1. Aquest és el motiu pel qual tenim dos dominis per les extensions 0 i 1.

Per això, durant la tercera setmana, la nostra tasca va consistir a adaptar adequadament el nou domini a les extensions 0 i 1 a més de preparar una bona documentació.

Modelat del domini

Extensió bàsica:

L'objectiu d'aquesta extensió és que el planner sigui capaç de trobar un pla per arribar a llegir tots els llibres objectiu, on cada llibre té 0 o 1 predecessor. També s'ha de complir que per tots els llibres del pla, tots els seus predecessors han d'haver sigut llegits en mesos anteriors.

Versió 1:

Tipus de Dades i Funcions:

- Hem definit dos tipus d'objectes: llibre i mes. Això ens permet organitzar la lectura en temporalment, relacionant cada llibre amb un mes concret.
- Utilitzem una funció posició per als mesos, indicant la posició de cada mes. Ens permet tractar les relacions entre els mesos posicionalment.

Predicats

- **(llegit ?x - llibre):** Aquest predicat s'utilitza per representar si un llibre ha estat llegit.
- **(llegit_en_mes ?x - llibre ?y - mes):** Aquest predicat relaciona un llibre amb un mes, indicant que el llibre x ha estat llegit en el mes y. Això és útil per organitzar la lectura temporalment, permetent-nos veure quan s'ha llegit cada llibre.
- **(llibre_desitjat ?x - llibre):** Es fa servir per marcar un llibre com a desitjat. Aquest predicat ens ajuda a identificar els llibres que encara volem llegir.
- **(predecessor ?x - llibre ?y - llibre):** Aquest predicat estableix una relació de precedència entre dos llibres. Si el llibre x és predecessor del llibre y, això implica que x hauria de ser llegit abans de y.
- **(llegit_abans ?x - llibre):** Indica si un llibre ha estat llegit en algun moment anterior, sense especificar quan. Això ens permet saber si un llibre ja ha estat llegit per l'usuari abans del pla.

Accions:

- **llibre_necessari:** Aquesta acció afegeix un llibre a la llista de llibres desitjats si aquest no està actualment desitjat i és predecessor d'un llibre desitjat.

- **llegir_llibre_amb_predecessor:** Permet la lectura d'un llibre desitjat en un mes específic si aquest no ha estat llegit abans i té un predecessor que ja ha estat llegit en un mes anterior.
- **llegir_llibre_sense_predecessor:** Afegeix al pla de lectura quan un llibre desitjat no té predecessor o el seu predecessor ja ha estat llegit abans del pla.

Versió 2:

Tipus de Dades i Funcions:

- Hem definit dos tipus d'objectes: llibre i mes. Això ens permet organitzar la lectura en temporalment, relacionant cada llibre amb un mes concret.

Predicats

- **(llegit ?x - llibre):** Aquest predicat s'utilitza per representar si un llibre ha estat llegit.
- **(llegint ?l - llibre):** Mostra si un llibre es llegeix en el mes actual.
- **(llegint_mes_anterior ?l - llibre):** Senyala si un llibre estava sent llegit en el mes anterior.
- **(predecessor ?x - llibre ?y - llibre):** Aquest predicat estableix una relació de precedència entre dos llibres. Si el llibre x és predecessor del llibre y, això implica que x hauria de ser llegit abans de y.
- **(mes_actual ?m - mes):** Identifica el mes actual dins del moment exacte de la planificació.
- **(mes_següent ?m1 - mes ?m2 - mes):** Estableix una relació entre dos mesos.

Accions:

- **llegir:** Permet començar a llegir un llibre en un mes específic. Les precondicions asseguren que el llibre no ha estat llegit ni està sent llegit actualment, i que no hi ha predecessors pendents de llegir. L'efecte és que el llibre comença a ser llegit.
- **avancar_mes:** Aquesta acció avança el sistema al mes següent. Canvia el mes actual i actualitza l'estat de lectura dels llibres: els que s'estaven llegint passen a haver estat llegits el mes anterior, i els que es van començar a llegir el mes anterior es marquen com a llegits.

Extensió 1:

L'objectiu d'aquesta extensió és que el planner sigui capaç de trobar un pla per arribar a llegir tots els llibres objectiu, on cada llibre té 0 o N predecessors, però cap de paral·lel. També s'ha de complir que per tots els llibres del pla, tots els seus predecessors han d'haver sigut llegits en mesos anteriors.

Versió 1:

Tipus de Dades i Funcions

- **Tipus:** Es mantenen els tipus llibre i mes, indicant que el sistema gestiona llibres i els relaciona amb els mesos de lectura.
- **Funcions:** La funció posició segueix present, per determinar l'ordre dels mesos.

Predicats

- **(llegit ?x - llibre):** Indica si un llibre ha estat llegit.
- **(llegit_en_mes ?x - llibre ?y - mes):** Relaciona un llibre amb el mes en què ha estat llegit.
- **(llibre_desitjat ?x - llibre):** Marca un llibre com a desitjat per a la lectura.
- **(predecessor ?x - llibre ?y - llibre):** Estableix una relació de precedència entre dos llibres.

Accions

- **llibre_necessari:** Afegeix un llibre a la llista de desitjats si no està actualment desitjat i és predecessor d'un llibre desitjat.
- **llegir:** Permet llegir un llibre desitjat en un mes específic. Les precondicions verifiquen que el llibre no hagi estat llegit, que tots els seus predecessors hagin estat llegits en mesos anteriors. L'efecte és que el llibre es marca com a llegit en aquell mes.

Versió 2:

Tipus de Dades i Funcions

- **Tipus:** llibre i mes, indicant que el sistema tracta sobre llibres i els relaciona amb els mesos en que són llegits.
- **Funcions:** La funció posicio s'utilitza per determinar l'ordre dels mesos.

Predicats

- **(llegit ?l - llibre):** Aquest predicat indica si un llibre ja ha estat llegit.
- **(llegint ?l - llibre):** Mostra si un llibre està sent llegit actualment.
- **(llegint_mes_anterior ?l - llibre):** Indica que un llibre es llegia en el mes anterior.
- **(mes_actual ?m - mes):** Identifica quin mes és l'actual dins del sistema.
- **(mes_següent ?m1 - mes ?m2 - mes):** Estableix una relació entre els mesos.
- **(predecessor ?lp - llibre ?p - llibre):** Defineix una relació de precedència entre dos llibres.

Accions

- **llegir:** Permet començar la lectura d'un llibre en un mes específic. Les precondicions assegurin que el llibre no ha estat llegit ni està sent llegit en el moment o el mes anterior, i que no hi ha predecessors pendents de lectura. L'efecte és que el llibre comença a ser llegit.
- **avancar_mes:** Aquesta acció avança el sistema al mes següent. Actualitza l'estat del mes actual i modifica els llibres que estaven en procés de lectura passen a ser considerats com llegits el mes anterior, i els que estaven llegint es marquen com a llegits.

Extensió 2:

L'objectiu d'aquesta extensió és que el planner sigui capaç de trobar un pla per arribar llegir tots els llibres objectiu, on cada llibre té 0 o N predecessors, a més a més pot tenir de 0 a M paral·lels. També s'ha de complir que per tots els llibres del pla, tots els seus predecessors han d'haver sigut llegits en mesos anteriors i els paral·lels en el mateix mes o en l'anterior (el següent per la relació simètrica).

Tipus de Dades

- **Tipus:** Es mantenen els tipus llibre i mes, indicant que el sistema gestiona llibres i els relaciona amb els mesos de lectura.

Predicats

- **(llegit ?l - llibre):** Indica si un llibre ja ha estat llegit.
- **(llegint ?l - llibre):** Mostra si un llibre està sent llegit en el moment actual.
- **(llegint_mes_anterior ?l - llibre):** Indica que un llibre es llegia en el mes anterior.
- **(mes_seguint ?m1 - mes ?m2 - mes):** Estableix una relació entre els mesos.
- **(mes_actual ?m - mes):** Identifica el mes actual dins del sistema.
- **(predecessor ?lp - llibre ?l1 - llibre):** Defineix una relació de precedència entre dos llibres.
- **(paralel ?l1 - llibre ?l2 - llibre):** Aquest nou predicat indica que dos llibres poden ser llegits de manera paral·lela.

Accions

- **llegir:** Permet començar la lectura d'un llibre en un mes determinat. Les precondicions asseguren que el llibre no està en procés de ser llegit, ja sigui en el mes actual o anterior, i que tots els seus predecessors hagin estat llegits. L'efecte és que el llibre comença a ser llegit.
- **avancar_de_mes:** Aquesta acció avança el sistema al mes següent. Les precondicions inclouen la comprovació que no hi hagi llibres llegits en paral·lel pendents de ser finalitzats el mes anterior. Els efectes inclouen l'actualització del mes actual i el canvi d'estat dels llibres de llegint_mes_anterior a llegit, i de llegint a llegint_mes_anterior.

Extensió 3:

L'objectiu d'aquesta extensió és que el planner sigui capaç de trobar un pla per arribar a llegir tots els llibres objectiu, on cada llibre té 0 o N predecessors, a més a més pot tenir de 0 a M paral·lels. També s'ha de complir que per tots els llibres del pla, tots els seus predecessors han d'haver sigut llegits en mesos anteriors i els paral·lels en el mateix mes o en l'anterior (el següent per la relació simètrica). I a més a més, el planner controla que no es llegeixin més de 800 pàgines al mes.

Tipus de Dades i Funcions

- **Tipus:** Es mantenen els tipus llibre i mes.
- **Funcions:** `pagines(?l - llibre)`: Representa el nombre total de pàgines d'un llibre. `pag_llegides(?m - mes)`: Aquesta funció serveix per guardar i actualitzar el nombre total de pàgines llegides en un mes determinat.

Predicats

- **(llegit ?l - llibre)**: Indica si un llibre ha estat llegit.
- **(llegint ?l - llibre)**: Mostra si un llibre està sent llegit actualment.
- **(llegint_mes_anterior ?l - llibre)**: Indica si un llibre va començar a ser llegit el mes anterior.
- **(mes_següent ?m1 - mes ?m2 - mes)**: Estableix una relació entre els mesos.
- **(mes_actual ?m - mes)**: Identifica el mes actual dins del sistema.
- **(predecessor ?lp - llibre ?l1 - llibre)**: Defineix una relació de precedència entre dos llibres.
- **(paralel ?l1 - llibre ?l2 - llibre)**: Indica que dos llibres poden ser llegits de manera paral·lela.

Accions

- **llegir**: Aquesta acció permet començar la lectura d'un llibre en un mes determinat. Les precondicions inclouen que el llibre no està en procés de ser llegit ni ha estat llegit, que tots els predecessors i paral·lels han estat llegits, i que la suma total de pàgines llegides en el mes actual més les pàgines del llibre actual no superi les 800. L'efecte és començar a llegir el llibre i incrementar el comptador de pàgines llegides d'aquest mes.
- **avancar_de_mes**: Aquesta acció avança el sistema al mes següent. Les precondicions i efectes són els mateixos a les versions anteriors del domini.

Modelat dels problemes

Al modelat dels problemes explicarem de manera general com hem plantejat els problemes, al següent apartat (el dels jocs de proves) tenim una descripció més detallada de cada un dels problemes.

Cada un dels nostres problemes consten de diversos elements.

1. Definició del domini

- **Objects:** Es declara un gran nombre de llibres que ens serviran llavors per fer les proves així com els mesos de gener a desembre.

2. Configuració inicial (Init)

- **Configuració dels mesos:** Es configuren les posicions dels mesos i s'estableixen les relacions pel nou domini (mes_següent x y). Cada una de les configuracions dels mesos servirà pel domini vell o nou respectivament.
- **Diferents proves:** Es configuren les diferents proves que ens acabaran servint per poder executar el problema. Definint els llibres predecessors, desitjats, llegits i paral·lels.
- **Nombre de pàgines:** S'assigna si escau, només en l'extensió 3, un nombre definit de pàgines per a cada un dels diferents llibres a llegir.

3. Objectiu (Goal):

- **Restricció general:** L'objectiu és assegurar que tots els llibres desitjats o llibres necessaris (aquells que són predecessors o paral·lels dels desitjats) acabaran sent llegits.

Conjunt de problemes de prova i resultats

Extensió 0 o bàsica problema 1:

Al ser el primer problema de l'extensió bàsica hem dedicat els nostres esforços i el joc de proves a comprovar els problemes que a priori són més trivials però que tot i així a nosaltres al principi ens fallaven. Com que són més trivials els problemes proposats al joc de proves n'hem fet uns quants.

Afegim una saga de llibres anomenada pilarsterra de la qual no ens hem llegit cap llibre. El llibre desitjat és el segon, com que el primer és predecessor del segon s'hauria de llegir el primer abans que el segon i el segon després. El tercer no cal que se'l llegeixi perquè no és ni desitjat ni predecessor.

Per les rondalles mallorquines fem que les desitjades siguin les rondalles 1, 2 i 5. S'hauran de llegir totes elles perquè per aconseguir llegir la rondalla 5, s'han d'haver llegit les seves predecessores.

Realitzem una prova pels llibres de poesia declarant que cap dels llibres de poesia ha estat llegit encara. S'estableix que només el primer llibre (poesia1) és desitjat per tant tot i haver tres llibres només s'haurà de llegir aquest.

Tenim una altra prova pel llibre Nesbo que al no haver estat llegit abans i al ser deistjat haurà de ser llegit.

Finalment, les dues últimes proves consisteixen en "mecanoscrit" que no ha estat llegit però no és desitjat per tant no s'ha de llegir i "Placa Diamant" que sí que és desitjat però que ha estat llegit anteriorment.

Hem comprovat les proves amb cada un dels dos dominis que tenim i ens resolen el problema de manera satisfactòria i així com esperàvem. A l'esquerra tenim la foto amb el domini antic i a la dreta la del nou:

| | |
|--|---------------------------------------|
| 0: LLIBRE_NECESSARI PILARSTERRA1 | 0: LLEGIR PILARSTERRA1 GENER |
| 1: LLEGIR_LLIBRE_SENSE_PREDECESOR PILARSTERRA1 GENER | 1: AVANCAR_MES GENER FEBRER |
| 2: LLEGIR_LLIBRE_SENSE_PREDECESOR RONDALLAMALLORQUINA1 GENER | 2: LLEGIR PILARSTERRA2 FEBRER |
| 3: LLEGIR_LLIBRE_AMB_PREDECESOR RONDALLAMALLORQUINA2 FEBRER | 3: LLEGIR NESBO FEBRER |
| 4: LLEGIR_LLIBRE_SENSE_PREDECESOR POESIA1 GENER | 4: LLEGIR POESIA1 FEBRER |
| 5: LLEGIR_LLIBRE_SENSE_PREDECESOR NESBO GENER | 5: LLEGIR RONDALLAMALLORQUINA1 FEBRER |
| 6: LLEGIR_LLIBRE_AMB_PREDECESOR PILARSTERRA2 DESEMBRE | 6: AVANCAR_MES FEBRER MAR |
| 7: LLIBRE_NECESSARI RONDALLAMALLORQUINA4 | 7: LLEGIR RONDALLAMALLORQUINA2 MAR |
| 8: LLIBRE_NECESSARI RONDALLAMALLORQUINA3 | 8: AVANCAR_MES MAR ABRIL |
| 9: LLEGIR_LLIBRE_AMB_PREDECESOR RONDALLAMALLORQUINA3 MAR | 9: LLEGIR RONDALLAMALLORQUINA3 ABRIL |
| 10: LLEGIR_LLIBRE_AMB_PREDECESOR RONDALLAMALLORQUINA4 ABRIL | 10: AVANCAR_MES ABRIL MAIG |
| 11: LLEGIR_LLIBRE_AMB_PREDECESOR RONDALLAMALLORQUINA5 DESEMBRE | 11: LLEGIR RONDALLAMALLORQUINA4 MAIG |
| 12: REACH-GOAL | 12: AVANCAR_MES MAIG JUNY |
| | 13: LLEGIR RONDALLAMALLORQUINA5 JUNY |
| | 14: AVANCAR_MES JUNY JULIOL |

Extensió 0 o bàsica problema 2:

Aquest segon problema presenta un escenari més complex que el primer. Fem cinc proves amb diferent complexitat.

En primer lloc, la primera prova declara tres llibres de manera que el primer és predecessor del segon i el segon del tercer. Tots tres són llibres desitjats per tant s'han de llegir els tres.

La segona prova consta de quatre llibres de harrypotter que són predecessors com abans però l'únic que és desitjat és l'últim i el primer ja ha estat llegit. Per tant, de fer-ho bé hauria de llegir el segon, tercer i quart llibre.

La tercera prova ens serveix per comprovar que en cap cas el pla ens fa llegir llibres innecessaris. Iniciem aquesta prova amb dos llibres de tintin (un predecessor de l'altre). Igualment al ser desitjat el primer només s'ha de llegir aquest, ignorant el segon.

La quarta prova és bastant senzilla, simplement afegim un llibre que ni té predecessors ni volem llegir per comprovar que efectivament no se'l llegeix.

Finalment, per a la cinquena prova afegim la saga de pilars de la terra, la qual té tres llibres predecessors de manera lineal i es vol llegir el segon. Per tant, hauria de llegir el primer i al tercer ignorar-lo.

L'objectiu global del problema és garantir que per a cada llibre, o bé no estigui marcat com a desitjat o bé hagi estat llegit. Si executem el codi amb cadascun dels dos dominis veurem que sempre es satisfà el nostre objectiu i per tant els nostres problemes fan la feina que haurien de fer. A l'esquerra tenim la foto amb el domini antic i a la dreta la del nou:

| | |
|--|-------------------------------|
| 0: LLIBRE_NECESSARI PILARSTERRA1 | 0: LLEGIR HARRYPOTTER2 GENER |
| 1: LLEGIR_LLIBRE_SENSE_PREDECESOR PILARSTERRA1 GENER | 1: AVANCAR_MES GENER FEBRER |
| 2: LLEGIR_LLIBRE_SENSE_PREDECESOR STARTREK1 GENER | 2: LLEGIR PILARSTERRA1 FEBRER |
| 3: LLEGIR_LLIBRE_AMB_PREDECESOR STARTREK2 FEBRER | 3: AVANCAR_MES FEBRER MAR |
| 4: LLEGIR_LLIBRE_SENSE_PREDECESOR TINTIN1 GENER | 4: LLEGIR HARRYPOTTER3 MAR |
| 5: LLEGIR_LLIBRE_AMB_PREDECESOR PILARSTERRA2 DESEMBRE | 5: AVANCAR_MES MAR ABRIL |
| 6: LLEGIR_LLIBRE_AMB_PREDECESOR STARTREK3 DESEMBRE | 6: LLEGIR PILARSTERRA2 ABRIL |
| 7: LLIBRE_NECESSARI HARRYPOTTER3 | 7: LLEGIR HARRYPOTTER4 ABRIL |
| 8: LLIBRE_NECESSARI HARRYPOTTER2 | 8: LLEGIR STARTREK1 ABRIL |
| 9: LLEGIR_LLIBRE_SENSE_PREDECESOR HARRYPOTTER2 GENER | 9: LLEGIR TINTIN1 ABRIL |
| 10: LLEGIR_LLIBRE_AMB_PREDECESOR HARRYPOTTER3 FEBRER | 10: AVANCAR_MES ABRIL MAIG |
| 11: LLEGIR_LLIBRE_AMB_PREDECESOR HARRYPOTTER4 DESEMBRE | 11: LLEGIR STARTREK2 MAIG |
| 12: REACH-GOAL | 12: AVANCAR_MES MAIG JUNY |
| | 13: LLEGIR STARTREK3 JUNY |
| | 14: AVANCAR_MES JUNY JULIOL |

Extensió 1 problema 1:

Aquest primer problema de l'extensió 1 introdueix un nivell de complexitat més elevat respecte als problemes anteriors. Presenta un parell de proves amb diverses condicions per a la lectura de llibres.

En la primera prova, s'estableix una relació entre llibres d'animals. Per llegir el llibre objectiu "reptils_expert" s'han d'haver llegit els seus llibres predecessors "serp" i "sargantana", i per llegir "serp" i "sargantana" s'han d'haver llegit els llibres de cada tipus d'animal de cada espècie ("sargantanapitiusa" o "cobra", per exemple). D'aquesta manera, es crea una estructura de dependències predecessores en forma d'arbre.

La segona prova se centra en l'objectiu de llegir llibre "malalties_expert" el qual està precedit per la lectura dels llibres de quatre malalties, dues de les quals ja han estat llegides anteriorment. Aquesta prova també està estructurada en forma d'arbre tot i que més senzill.

L'objectiu global del problema torna a ser garantir que per a cada llibre, o bé no estigui marcat com a desitjat o bé hagi estat llegit. Si executem el codi amb cadascun dels dos dominis veurem que sempre se satisfà el nostre objectiu i, per tant, els nostres problemes fan la feina que haurien de fer. A l'esquerra tenim la foto amb el domini antic i a la dreta la del nou:

| | |
|--|----------------------------------|
| 0: LLIBRE_NECESSARI MALARIA | |
| 1: LLEGIR MALARIA GENER | |
| 2: LLIBRE_NECESSARI HEPATITIS | |
| 3: LLEGIR HEPATITIS GENER | |
| 4: LLEGIR MALALTIES_EXPERT FEBRER | 0: LLEGIR PITONS GENER |
| 5: LLIBRE_NECESSARI TORTUGA | 1: LLEGIR SERPBLANCA GENER |
| 6: LLEGIR TORTUGA GENER | 2: LLEGIR LLANGARDAIX GENER |
| 7: LLIBRE_NECESSARI SARGANTANA | 3: LLEGIR HEPATITIS GENER |
| 8: LLIBRE_NECESSARI LLANGARDAIX | 4: LLEGIR TORTUGA GENER |
| 9: LLEGIR LLANGARDAIX GENER | 5: AVANCAR_MES GENER FEBRER |
| 10: LLIBRE_NECESSARI SARGANTANAPITIUSA | 6: LLEGIR ESCURCO FEBRER |
| 11: LLEGIR SARGANTANAPITIUSA GENER | 7: AVANCAR_MES FEBRER MAR |
| 12: LLEGIR SARGANTANA FEBRER | 8: LLEGIR SARGANTANAPITIUSA MAR |
| 13: LLIBRE_NECESSARI SERP | 9: AVANCAR_MES MAR ABRIL |
| 14: LLIBRE_NECESSARI PITONS | 10: LLEGIR MALARIA ABRIL |
| 15: LLEGIR PITONS GENER | 11: AVANCAR_MES ABRIL MAIG |
| 16: LLIBRE_NECESSARI SERPBLANCA | 12: LLEGIR SERP MAIG |
| 17: LLEGIR SERPBLANCA GENER | 13: LLEGIR SARGANTANA MAIG |
| 18: LLIBRE_NECESSARI ESCURCO | 14: AVANCAR_MES MAIG JUNY |
| 19: LLEGIR ESCURCO GENER | 15: LLEGIR MALALTIES_EXPERT JUNY |
| 20: LLEGIR SERP FEBRER | 16: LLEGIR REPTILS_EXPERT JUNY |
| 21: LLEGIR REPTILS_EXPERT MAR | 17: AVANCAR_MES JUNY JULIOL |
| 22: REACH-GOAL | |

Extensió 1 problema 2:

Aquest segon problema de l'extensió 1 a nivell de conceptes és com el primer però molt més complex i gran. Si ens hi fixem és un arbre de llibres sobre dispositius. El llibre objectiu és dispositius, el qual per poder ser llegit necessita que ens haguem llegit telèfons, portàtils i televisors i, aquests per ser llegits necessiten cadascun dels dispositius que hagin sigut llegits abans. Si ens hi fixem els llibres dels dispositius d'Apple ja han sigut llegits anteriorment.

Un cop més, l'objectiu del problema torna a ser el mateix i veiem com amb ambdós dominis funciona. A l'esquerra tenim la foto amb el domini antic i a la dreta la del nou:

| | |
|----------------------------------|------------------------------|
| 0: LLIBRE_NECESSARI TELEVISORS | 0: LLEGIR GALAXY GENER |
| 1: LLIBRE_NECESSARI SAMSUNGTV | 1: LLEGIR ZENBOOK GENER |
| 2: LLEGIR SAMSUNGTV GENER | 2: LLEGIR SAMSUNGTV GENER |
| 3: LLIBRE_NECESSARI GOOGLETV | 3: AVANCAR_MES GENER FEBRER |
| 4: LLEGIR GOOGLETV GENER | 4: LLEGIR HUAWEIPHONE FEBRER |
| 5: LLEGIR TELEVISORS FEBRER | 5: AVANCAR_MES FEBRER MAR |
| 6: LLIBRE_NECESSARI PORTATILS | 6: LLEGIR CHROMEBOOK MAR |
| 7: LLIBRE_NECESSARI ZENBOOK | 7: LLEGIR GOOGLETV MAR |
| 8: LLEGIR ZENBOOK GENER | 8: AVANCAR_MES MAR ABRIL |
| 9: LLIBRE_NECESSARI CHROMEBOOK | 9: LLEGIR TELEFONS ABRIL |
| 10: LLEGIR CHROMEBOOK GENER | 10: LLEGIR PORTATILS ABRIL |
| 11: LLEGIR PORTATILS FEBRER | 11: LLEGIR TELEVISORS ABRIL |
| 12: LLIBRE_NECESSARI TELEFONS | 12: AVANCAR_MES ABRIL MAIG |
| 13: LLIBRE_NECESSARI GALAXY | 13: LLEGIR DISPOSITIUS MAIG |
| 14: LLEGIR GALAXY GENER | 14: AVANCAR_MES MAIG JUNY |
| 15: LLIBRE_NECESSARI HUAWEIPHONE | |
| 16: LLEGIR HUAWEIPHONE GENER | |
| 17: LLEGIR TELEFONS FEBRER | |
| 18: LLEGIR DISPOSITIUS MAR | |
| 19: REACH-GOAL | |

Extensió 2 problema 1:

Aquest problema és el senzill dels dos de l'extensió 2. Estableix una relació de predecessors entre tots els cinc llibres d'Agatha Christie i una altra relació de predecessors entre els tres de Ken Follet. A més a més, s'estableixen relacions paral·leles entre els llibres kenfollet1 i agathachristie5 de manera bidireccional cosa que obliga a que siguin llegits al mateix mes o amb màxim un més de diferència. Indiquem que el llibre agathachristie1 ha sigut llegit abans per tant no cal que torni a ser llegit. A més a més, indiquem que els llibres desitjats són agathachristie4 i kenfollet3.

L'objectiu final torna a ser el de sempre i veiem més a baix com funciona amb l'únic domini que tenim en aquest cas.

0: LLEGIR AGATHACHRISTIE2 GENER
1: AVANCAR_DE_MES GENER FEBRER
2: LLEGIR AGATHACHRISTIE3 FEBRER
3: AVANCAR_DE_MES FEBRER MAR
4: LLEGIR AGATHACHRISTIE4 MAR
5: LLEGIR KENFOLLET1 MAR
6: AVANCAR_DE_MES MAR ABRIL
7: LLEGIR KENFOLLET2 ABRIL
8: LLEGIR AGATHACHRISTIE5 ABRIL
9: AVANCAR_DE_MES ABRIL MAIG
10: LLEGIR KENFOLLET3 MAIG
11: AVANCAR_DE_MES MAIG JUNY
12: AVANCAR_DE_MES JUNY JULIOL

Extensió 2 problema 2:

Aquest joc de proves és el més complex que tenim. A part de perquè provi l'extensió també l'hem fet per **avaluar el temps que trigava el nostre domini a resoldre'l** que no ha estat pas gaire dolent (uns 7 segons).

El joc de proves es basa en l'anàlisi de llibres al títol dels quals hi consti algun nom de poblacions dels Països Catalans, estructurats de manera complexa i interrelacionada.

S'han definit múltiples relacions entre els llibres de pobles (com Figueres), províncies (com Girona) i països (com Catalunya). Les dependències entre els elements del problema es distribueixen a través d'aquestes tres capes, i al ser desitjats els llibres dels tres països dels Països Catalans necessitem que s'hagin llegit tots els llibres anteriors (ja que tots són predecessors directes o indirectes d'aquests tres). S'estableixen relacions de predecessors entre ciutats/pobles, províncies i països, creant una xarxa complexa d'interdependències.

A més a més, hem creat també relacions paral·leles entre els llibres de certes ciutats, províncies i països, com ara entre "catalunya" i "arta", forçant que es llegeixin en el mateix mes o amb màxim d'un mes de diferència.

L'objectiu principal d'aquest problema és comprovar el temps d'execució així com comprovar que el nostre domini funciona correctament, cosa que fa, com es pot observar a continuació:


```

0: AVANCAR_DE_MES GENER FEBRER
1: LLEGIR FIGUERES FEBRER
2: LLEGIR CENTELLES FEBRER
3: LLEGIR FERRERIES FEBRER
4: LLEGIR EIVISSA FEBRER
5: LLEGIR ALACANT FEBRER
6: LLEGIR CASTELLO FEBRER
7: LLEGIR XATIVA FEBRER
8: LLEGIR PROVINCIA TARRAGONA FEBRER
9: LLEGIR SOLSONA FEBRER
10: AVANCAR_DE_MES FEBRER MAR
11: LLEGIR PROVINCIAL LEIDA MAR
12: LLEGIR BARCELONA MAR
13: AVANCAR_DE_MES MAR ABRIL
14: LLEGIR PROVINCIA BARCELONA ABRIL
15: LLEGIR GIRONA ABRIL
16: AVANCAR_DE_MES ABRIL MAIG
17: LLEGIR PROVINCIA GIRONA MAIG
18: AVANCAR_DE_MES MAIG JUNY
19: LLEGIR CATALUNYA JUNY
20: LLEGIR SANTANTONI JUNY
21: LLEGIR PALMA JUNY
22: LLEGIR SAGUNT JUNY
23: AVANCAR_DE_MES JUNY JULIOL
24: LLEGIR PROVINCIA CASTELLO JULIOL
25: LLEGIR PROVINCIA MALLORCA JULIOL
26: LLEGIR PROVINCIA EIVISSA JULIOL
27: LLEGIR MAO JULIOL
28: LLEGIR GANDIA JULIOL
29: AVANCAR_DE_MES JULIOL AGOST
30: LLEGIR PROVINCIA ALACANT AGOST
31: LLEGIR PROVINCIA VALENCIA AGOST
32: LLEGIR PROVINCIA MENORCA AGOST
33: AVANCAR_DE_MES AGOST SETEMBRE
34: LLEGIR ILLES BALEARS SETEMBRE
35: LLEGIR PAIS VALENCIA SETEMBRE
36: AVANCAR_DE_MES SETEMBRE OCTUBRE
37: AVANCAR_DE_MES OCTUBRE NOVEMBRE

```

Extensió 3 problema 1:

Aquest problema no té molt misteri perquè és la còpia d'un problema dels anteriors (extensió 2 problema 1) però introduïnt un nou component relacionat amb les pàgines llegides. Cada llibre té una quantitat específica de pàgines i màxim es poden llegir 800 pàgines al mes.

Veiem com funciona perfectament perquè per exemple, kenfollet3 que té 799 se'l llegeix en un mes de manera independent.

```

0: LLEGIR AGATHACHRISTIE2 GENER
1: AVANCAR_DE_MES GENER FEBRER
2: LLEGIR AGATHACHRISTIE3 FEBRER
3: AVANCAR_DE_MES FEBRER MAR
4: LLEGIR AGATHACHRISTIE4 MAR
5: AVANCAR_DE_MES MAR ABRIL
6: LLEGIR KENFOLLET1 ABRIL
7: LLEGIR AGATHACHRISTIE5 ABRIL
8: AVANCAR_DE_MES ABRIL MAIG
9: LLEGIR KENFOLLET2 MAIG
10: AVANCAR_DE_MES MAIG JUNY
11: LLEGIR KENFOLLET3 JUNY
12: AVANCAR_DE_MES JUNY JULIOL
13: AVANCAR_DE_MES JULIOL AGOST

```

Extensió 3 problema 2:

Aquest també és una adaptació d'un problema anterior (l'1 de la extensió bàsica). He mantingut les mateixes relacions de predecessors afegint un nombre de pàgines per a cada llibre així com algunes relacions paral·leles entre llibres. Un cop més, cada llibre té una quantitat específica de pàgines i màxim es poden llegir 800 pàgines al mes. L'objectiu és llegir tots els llibres desitjats que no han estat llegits anteriorment. A més a més caldrà llegir els llibres necessaris per llegir els llibres desitjats.

```
0: LLEGIR RONDALLAMALLORQUINA1 GENER
1: LLEGIR PILARSTERRA1 GENER
2: AVANCAR_DE_MES GENER FEBRER
3: LLEGIR RONDALLAMALLORQUINA2 FEBRER
4: AVANCAR_DE_MES FEBRER MAR
5: LLEGIR RONDALLAMALLORQUINA3 MAR
6: AVANCAR_DE_MES MAR ABRIL
7: LLEGIR RONDALLAMALLORQUINA4 ABRIL
8: AVANCAR_DE_MES ABRIL MAIG
9: LLEGIR RONDALLAMALLORQUINA5 MAIG
10: LLEGIR PILARSTERRA2 MAIG
11: AVANCAR_DE_MES MAIG JUNY
12: LLEGIR NESBO JUNY
13: AVANCAR_DE_MES JUNY JULIOL
14: LLEGIR POESIA1 JULIOL
15: AVANCAR_DE_MES JULIOL AGOST
16: AVANCAR_DE_MES AGOST SETEMBRE
```

Generador de problemas

A més a més del codi PDDL proporcionat, hem creat també dos arxius python (un per l'extensió 2 i un altre per la 3) per generar problemes aleatoris sobre els quals dur a terme experiments. Expliquem com els hem creat i com funcionen a continuació:

Generador per l'extensió 2

Podem trobar annexat en la entrega aquest generador, el codi del qual es pot dividir en les següents parts que expliquem a continuació:

- **Paràmetres d'entrada:** El codi rep diversos paràmetres des de l'entrada del sistema (mitjançant `sys.argv`). Aquests paràmetres s'utilitzen per definir les característiques del problema i són els següents:
 1. Nombre total de llibres del problema
 2. Nombre desitjat de llibres a llegir
 3. Nombre de llibres llegits prèviament
 4. Nombre de relacions de paral·lelisme entre llibres
 5. Densitat de predecessors (en una escala de l'1 al 10)
- **Creació del fitxer PDDL:** S'obre un fitxer en mode escriptura per escriure la definició del problema PDDL, el nom del qual es descriu amb els paràmetres anteriors.
- **Creació del graf:** S'utilitza `nx.gnp_random_graph` de la llibreria NetworkX per crear un graf dirigit aleatori amb una certa densitat de connexions entre els llibres, definida pel paràmetre d'entrada *Densitat de predecessors*.
- **Definició dels objectes i constants en el domini del problema:** Es defineixen els objectes, com ara els llibres i els mesos, que formen part del domini del problema. Els llibres es representen com a nodes del graf.
- **Inicialització de les relacions temporals:** Es defineixen les relacions temporals entre els mesos.
- **Establiment de relacions de paral·lelisme:** Es determinen i escriuen les relacions de paral·lelisme entre els llibres, indicant quins llibres poden són paral·lels entre ells.
- **Establiment de relacions de predecessors:** Es defineixen les relacions de predecessors entre els llibres.

- **Marcatge de llibres llegits i desitjats:** S'indica quins llibres s'han llegit i quins llibres es desitja llegir en el problema de planificació.
- **Definició de l'objectiu del problema:** Es defineix l'objectiu del problema de planificació, que en aquest cas és assegurar que tots els llibres desitjats siguin llegits.
- **Tancament del fitxer PDDL:** Es tanca el fitxer una vegada s'ha escrit tota la informació rellevant.

En resum, aquest codi crea un fitxer en llenguatge PDDL que descriu un problema de planificació pel domini de l'extensió 2 que ja hem explicat i que podem trobar annex en la entrega.

Generador per l'extensió 3

El generador de problemes per a l'extensió 3 és molt semblant al de l'extensió 2 pero afegeix una part nova després de la inicialització de les relacions temporals:

- **Assignació de pàgines als llibres:** S'assigna un nombre aleatori de pàgines (entre 50 i 799) a cada llibre del graf i s'escriu aquesta informació en el fitxer PDDL.

Utilització

Per utilitzar el generador hem de fer una crida al terminal de l'extensió del generador que vulguem utilitzar acompanyat dels següents arguments definits a continuació en el mateix ordre:

1. Nombre total de llibres del problema
2. Nombre desitjat de llibres a llegir
3. Nombre de llibres llegits prèviament
4. Nombre de relacions de paral·lelisme entre llibres
5. Densitat de predecessors (en una escala de l'1 al 10)

Així, per exemple, si volem generar un problema per l'extensió 3 amb 5 llibres, 2 dels quals desitjats, 1 llegit anteriorment, 1 relació de paral·lelisme i una densitat de predecessors de 8 sobre 10, hem de fer la següent crida al terminal:

```
\ python generador3.py 5 2 1 1 5
```

Experimentació

Experiment 1

Determinar quin de les dos versions de domini de l'extensió 0 és més eficient.

Observació i plantejament: L'Experiment 1 busca trobar quina de les dos versions de domini de la extensió 0 és més eficient. Aquest experiment es realitza executant per cada domini els dos problemes que tenim creats per l'extensió 0. Cal tenir en compte que les dues version arriben al mateix resultat per cada problema.

Hipòtesi:

- **Hipòtesi Nul·la (Ho):** Els dominis es comporten de la mateixa manera quan s'executa el mateix problema.
- **Hipòtesi Alternativa:** Un dels dos dominis té menor temps d'execució, genera menys estats i/o instància menys accions.

Mètode:

- **Elecció dels subjectes:** Per cada domini executarem els dos problemes que hem creat, el problema és el mateix.
- **Procediment:** Executarem cada domini amb els dos problemes i compararem resultats

Resultats:

Pel primer problema tenim els següents resultats:

Versió 1:

0.00 seconds instantiating 72 easy, 532 hard action templates

0.00 seconds reachability analysis, yielding 137 facts and 283 actions

0.00 seconds creating final representation with 122 relevant facts, 0 relevant fluents

0.00 seconds computing LNF

0.00 seconds building connectivity graph

0.00 seconds searching, evaluating 91 states, to a max depth of 0

0.00 seconds total time

Versió 2:

0.00 seconds instantiating 11 easy, 252 hard action templates
0.00 seconds reachability analysis, yielding 94 facts and 263 actions
0.00 seconds creating final representation with 92 relevant facts, 0 relevant fluents
0.00 seconds computing LNF
0.00 seconds building connectivity graph
0.00 seconds searching, evaluating 67 states, to a max depth of 0
0.00 seconds total time

I pel segon:

Versió 1:

0.00 seconds instantiating 0 easy, 537 hard action templates
0.00 seconds reachability analysis, yielding 96 facts and 159 actions
0.00 seconds creating final representation with 84 relevant facts, 0 relevant fluents
0.00 seconds computing LNF
0.00 seconds building connectivity graph
0.00 seconds searching, evaluating 82 states, to a max depth of 0
0.00 seconds total time

Versió 2:

0.00 seconds instantiating 11 easy, 228 hard action templates
0.00 seconds reachability analysis, yielding 88 facts and 239 actions
0.00 seconds creating final representation with 86 relevant facts, 0 relevant fluents
0.00 seconds computing LNF
0.00 seconds building connectivity graph
0.00 seconds searching, evaluating 64 states, to a max depth of 0
0.00 seconds total time

Conclusions:

Veiem que tot i que el temps d'execució sigui 0 en tots els casos, la versió 2 és més eficient, instància menys accions, i evalua menys estats ja que les accions són més directes, tot i que en els dos casos la profunditat és 0. A més a més, fa anar el fluent posició i gestiona els mesos de forma més senzilla i es fan menys exists i no s'utilitzen operadors de comparació.

Experiment 2

Determinar quin de les dos versions de domini de l'extensió 1 és més eficient.

Observació i plantejament: L'Experiment 2 busca trobar quina de les dos versions de domini de la extensió 1 és més eficient. Aquest experiment es realitza executant per cada domini els dos problemes que tenim creats per l'extensió 1. Cal tenir en compte que les dues versions arriben al mateix resultat per cada problema.

Hipòtesi:

- **Hipòtesi Nul·la (H₀):** Els dominis es comporten de la mateixa manera quan s'executa el mateix problema.
- **Hipòtesi Alternativa:** Un dels dos dominis té menor temps d'execució, genera menys estats i/o instància menys accions.

Mètode:

- **Elecció dels subjectes:** Per cada domini executarem els dos problemes que hem creat, el problema és el mateix.
- **Procediment:** Executarem cada domini amb els dos problemes i compararem resultats

Resultats:

Pel primer problema tenim els següents resultats:

Versió 1:

0.00 seconds instantiating 0 easy, 157 hard action templates
0.00 seconds reachability analysis, yielding 379 facts and 1181 actions
0.00 seconds creating final representation with 339 relevant facts, 0 relevant fluents
0.15 seconds computing LNF
0.00 seconds building connectivity graph
0.00 seconds searching, evaluating 241 states, to a max depth of 0
0.15 seconds total time

Versió 2:

0.00 seconds instantiating 11 easy, 384 hard action templates
0.00 seconds reachability analysis, yielding 96 facts and 395 actions
0.00 seconds creating final representation with 90 relevant facts, 0 relevant fluents
0.00 seconds computing LNF
0.00 seconds building connectivity graph

0.00 seconds searching, evaluating 102 states, to a max depth of 0
0.00 seconds total time

I pel segon:

Versió 1:

0.00 seconds instantiating 0 easy, 132 hard action templates
0.00 seconds reachability analysis, yielding 324 facts and 644 actions
0.00 seconds creating final representation with 285 relevant facts, 0 relevant fluents
0.00 seconds computing LNF
0.00 seconds building connectivity graph
0.00 seconds searching, evaluating 189 states, to a max depth of 0
0.00 seconds total time

Versió 2:

0.00 seconds instantiating 11 easy, 312 hard action templates
0.00 seconds reachability analysis, yielding 84 facts and 323 actions
0.00 seconds creating final representation with 78 relevant facts, 0 relevant fluents
0.00 seconds computing LNF
0.00 seconds building connectivity graph
0.00 seconds searching, evaluating 142 states, to a max depth of 0
0.00 seconds total time

Conclusions:

Veiem que el temps d'execució de la versió 1 pel problema 1 és l'únic que no és 0, a més a més, en els dos casos la versió 1 evalua més estats, en les dos versions la profunditat és 0, tot i que la versió 2 instància més accions. La conclusió és que la versió 2 segueix sent més eficient ja que les accions són més directes. A més a més, fa anar el fluent posició i gestiona els mesos de forma més senzilla i es fan menys exists i no s'utilitzen operadors de comparació.

Experiment 3

Determinar com afecta el nombre total de llibres en el temps total d'execució de problemes

Observació i plantejament: L'Experiment 3 busca observar com evoluciona el temps total d'execució amb l'augment del nombre total de llibres. Usarem el generador de problemes en l'extensió 2 per veure l'evolució dels temps d'execució.

Hipòtesi:

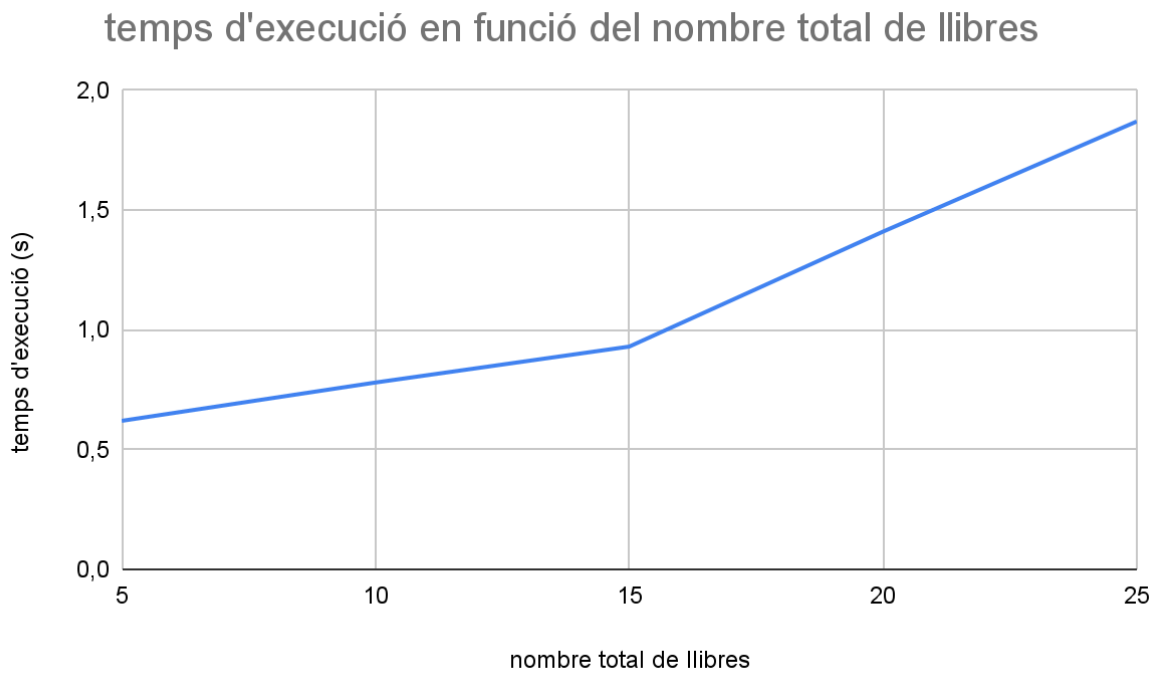
- **Hipòtesi Nul·la (Ho):** El temps d'execució no varia en funció del nombre de llibres totals del problema.
- **Hipòtesi Alternativa:** El temps d'execució es proporcional al nombre de llibres totals del problema.

Mètode:

- **Procediment:** Per al domini 2 executarem els problema amb 5, 10, 15, 20 i 25 llibres, tots amb els mateixos paràmetres: 2 desitjats, 1 llegit, 1 paral·lelisme, 5 de densitat predecessors
-

Resultats:

| <i>X llibres</i> | |
|--|-----------------------------------|
| <i>2 desitjats, 1 llegit, 1 paral·lelisme, 5 de densitat predecessors</i> | |
| nombre total de llibres del problema | temps total d'execució (s) |
| 5 | 0,62 |
| 10 | 0,78 |
| 15 | 0,93 |
| 20 | 1,41 |
| 25 | 1,87 |



Conclusions:

Veiem que, com es predeia en la hipòtesi alternativa, el fet d'augmentar el nombre total de llibres augmenta el temps d'execució. L'augment, però, no es del tot proporcional, més bé, és l'inici d'un creixement exponencial. Així, podem concloure que el temps d'execució creix exponencialment al augmentar el nombre de llibres.

Experiment 4

Com afecta el nombre de relacions de paral·lelisme en el temps total d'execució de problemes

Observació i plantejament: L'Experiment 4 busca observar com evoluciona el temps total d'execució amb l'augment de relacions de paral·lelisme entre llibres. Usarem el generador de problemes en l'extensió 2 per veure l'evolució dels temps d'execució.

Hipòtesi:

- **Hipòtesi Nul·la (Ho):** El temps d'execució no varia en funció del nombre de relacions de paral·lelisme del problema.
- **Hipòtesi Alternativa:** El temps d'execució es proporcional al nombre de relacions de paral·lelisme del problema.

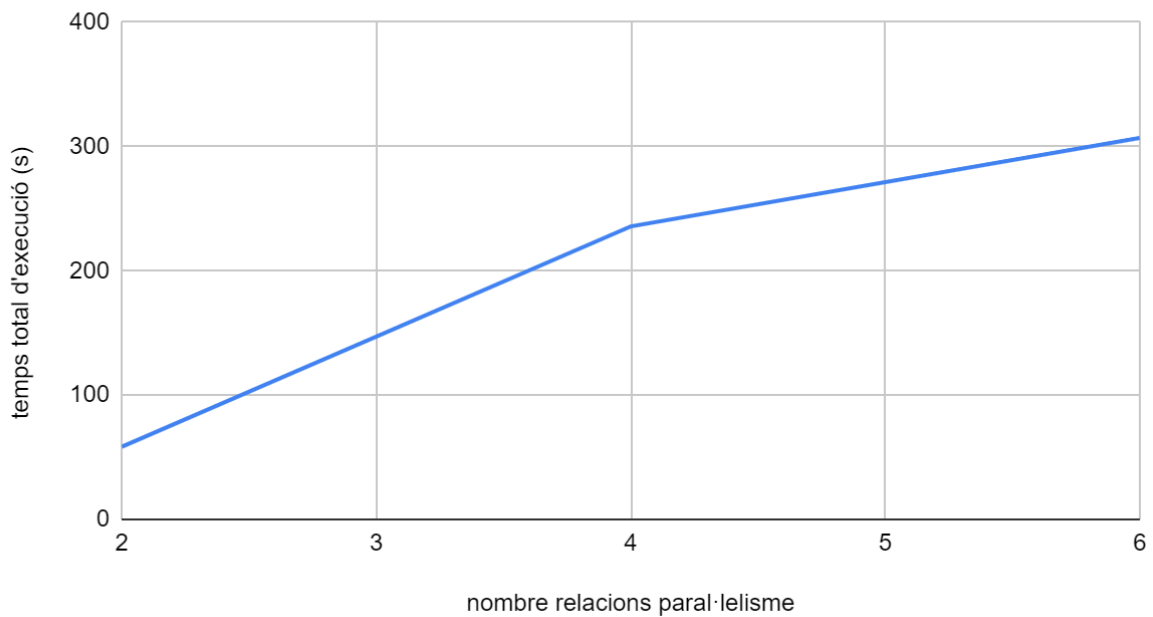
Mètode:

- **Procediment:** Per al domini 2 executarem els problema amb 2, 4 i 6 relacions de paral·lelisme, mantenint tots els altres paràmetres amb valors constants: 10 llibres, 3 desitjats, 0 llegits abans i una densitat de predecesors de 3 sobre 10.

Resultats:

| <i>X relacions de paral·lelisme</i> <i>10 llibres, 3 desitjats, 0 llegits, 3 de densitat predecessors</i> | |
|--|-----------------------------------|
| nombre relacions paral·lelisme | temps total d'execució (s) |
| 2 | 58,37 |
| 4 | 235,78 |
| 6 | 306,93 |

temps total d'execució (s) i nombre relacions paral·lelisme



Conclusions:

Veiem que, com es predeia en la hipòtesi alternativa, el fet d'augmentar el les relacions de paral·lelisme de llibres augmenta el temps d'execució. L'augment sembla ser logarítmic. Així, podem concloure que el temps d'execució creix logarítmicament al augmentar les relacions de paral·lelisme en el problema.

Experiment 5

Com afecta la densitat de predecessors en el temps total d'execució de problemes

Observació i plantejament: L'Experiment 5 busca observar com evoluciona el temps total d'execució amb l'augment de la densitat de llibres predecessors. Usarem el generador de problemes en l'extensió 3 per veure l'evolució dels temps d'execució.

Hipòtesi:

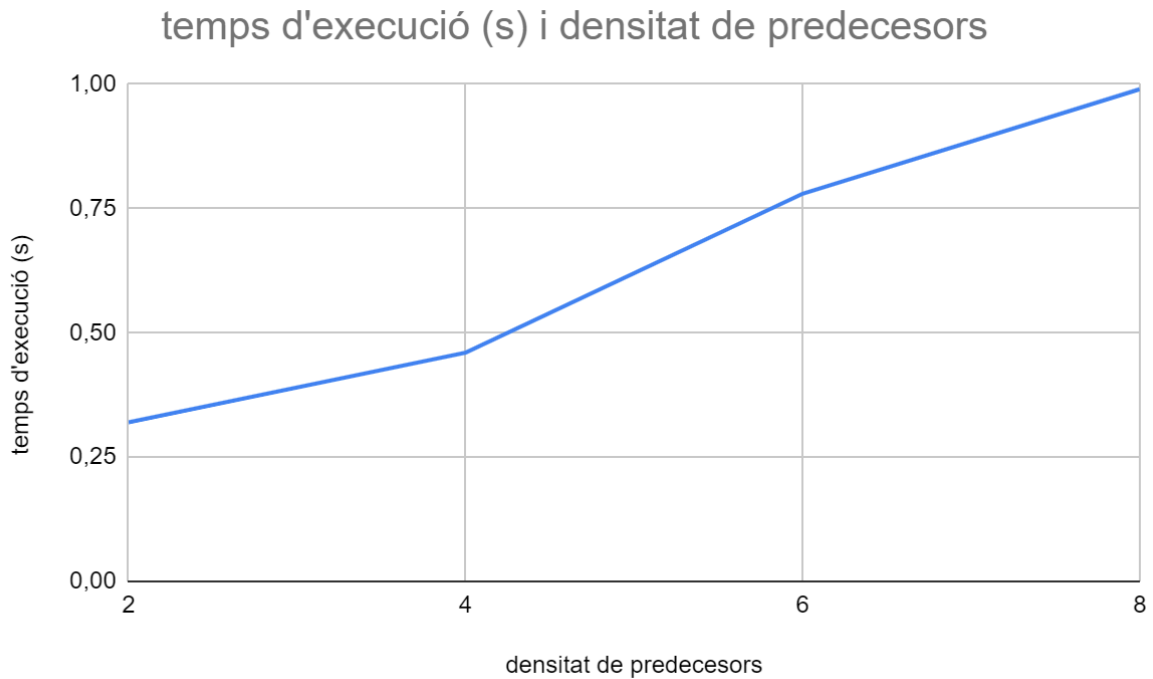
- **Hipòtesi Nul·la (Ho):** El temps d'execució no varia en funció de la densitat de predecessors del problema.
- **Hipòtesi Alternativa:** El temps d'execució es proporcional a la densitat de predecessors del problema.

Mètode:

- **Procediment:** Per al domini 3 executarem els problema amb 2, 4, 6 i 8 densitats de predecessors, mantenint tots els altres paràmetres amb valors constants: 6 llibres, 2 desitjats, 1 llegits abans i 1 relació de paral·lelisme entre llibres.

Resultats:

| X densitat de predecessors | |
|--|-----------------------------|
| <i>6 llibres, 2 desitjats, 1 llegits abans i 1 relació de paral·lelisme</i> | |
| densitat de predecesors | temps d'execució (s) |
| 2 | 0,32 |
| 4 | 0,46 |
| 6 | 0,78 |
| 8 | 0,99 |



Conclusions:

Veiem que, com es predeia en la hipòtesi alternativa, el fet d'augmentar el nombre total de llibres augmenta el temps d'execució. L'augment sembla ser proporcional. Així, podem concloure que el temps d'execució creix proporcionalment al augmentar la densitat de llibres predecesors en el problema.

Conclusions i satisfacció general

En conclusió, aquest treball ens ha ajudat a entendre el funcionament de planificadors, en aquest cas de lectura, així com del llenguatge PDDL, sobre el qual hem pogut treballar.

En la fase de creació del diferents dominis, ens vam adonar dels problemes que suposava el primer domini sobre el qual estavem treballar, i es per això que vam decidir canviar al segon model de domini, que a resultat ser més eficient, com ha demostrat la fase d'experimentació, en el experiments 1 i 2, a més a més de permetrens poder treballar sobre les extensions 2 i 3.

Els problemes que hem creat ens han permés comprovar que els diferens dominis funcionen correctament en diferents casos i casuístiques diverses, sovint complexes.

En la fase d'experimentació, finalment hem pogut observar, gràcies al generador de problemes aleatori que hem creat, com els temps d'execució augmenten de diferents formes en funció del nombre total de llibres, de les relacions de paral·lelisme i de la densitat de predecessors.