

Exercicis de planificació**Exercici 1. Qüestions.**

Respongueu a les qüestions següents en aquest mateix full:

1. Un planificador permet preveure errors mentre que un controlador segueix un comportament pre-programat.

- ☐ Cert.
☐ Fals.
☐ Depèn.

2. Els beneficis de la planificació en la resolució de problemes són

- ☐ La reducció del problema per emprar heurístiques
☐ La facilitat de ser independent del domini, depenent de les heurístiques que s'hi apliquin.
☐ La reducció de la cerca, la resolució de conflictes entre objectius i la recuperació davant d'errors

3. Indica tres mecanismes per afrontar un conflicte entre una acció i un enllaç causal

- ☐
☐
☐

4. Indica la diferència entre una acció i un operador.

- ☐

5. Una mesura d'utilitat de la planificació permet

- ☐ Conèixer quina acció realitzar a continuació.
☐ Avaluar la conveniència de canviar l'estratègia de refinament d'un pla
☐ Decidir quan planificar i quan actuar.

6. La planificació condicional permet elaborar plans en moltes més circumstàncies que altres mètodes clàssics. Explica per què.

- ☐

7. L'ús de sensors és necessari perquè

- ☐ Permeten identificar un objecte desitjat en el món.
☐ Garanteixen la possibilitat de canviar el món per a què els objectes s'adeqüin a les descripcions cercades.
☐ Permeten rastrejar el món.

8. Anomena tres aplicacions de la planificació a la fabricació de productes.

- ☐
☐
☐

Exercici 2. Assemblatge de components.

Es planteja el següent problema: com fer que un robot assembli els tres components que componen una màquina.

Operacions disponibles:

- Es pot assemblar el component 1, sinó està ja assemblat.
- Es pot assemblar el component 2, si el component 1 ja està assemblat. En aquest cas, el component 1 quedarà desassemblat.
- Es pot assemblar el component 2, si el component 1 no està assemblat.
- Es pot assemblar el component 3, si els components 1 i 2 ja han estat assemblats.

Estat inicial: Tenim tres components a assemblar: l'1, el 2 i el 3.

Objectiu: Tots els components han de quedar assemblats

Definiu:

- Els operadors
- L'estat inicial i l'objectiu
- Aplicar un mètode de progressió (cerca en amplitud) per resoldre el problema.
- Aplicar alguna heurística per millorar els resultats
- Aplicar un mètode de regressió.

En totes les solucions es demana fer el seguiment de l'algorisme mitjançant l'arbre de cerca generat que us servirà per comparar els resultats.

Exercici 3. El món dels blocs.

Donat el següent problema del món dels blocs, proporcionar les traces de solució que produirien un planificador lineal (tipus STRIPS) i un planificador no lineal (tipus TWEAK).

Predicats:

SOBRE(x,y): El bloc x està sobre el bloc y.
 SOBRELATAULA(x): El bloc x està sobre la taula
 LLIURE(x): No hi ha res sobre el bloc x
 AGAFAT(x): El braç té agafat el bloc x
 BRAÇLLIURE: El braç no està agafant cap bloc.

Estat Inicial:

SOBRE(C,D), SOBRE(A,B), SOBRELATAULA(B), SOBRELATAULA(D), BRAÇLLIURE, LLIURE(A), LLIURE(C)

Estat Final:

SOBRE(C,B), SOBRE(D,A), SOBRELATAULA(B), SOBRELATAULA(A), BRAÇLLIURE

Operadors disponibles:

APILAR(x,y)
 Prec.: LLIURE(y), AGAFAT(x)
 Efectes: NO LLIURE(y), NO AGAFAT(x)
 BRAÇLLIURE, SOBRE(x,y)

DESAPILAR(x,y)
 Prec.: SOBRE(x,y), LLIURE(x), BRAÇLLIURE
 Efectes: NO SOBRE(x,y), NO BRAÇLLIURE
 AGAFAT(x), LLIURE(y)

AGAFAR(x)
 Prec.: LLIURE(x), SOBRELATAULA(x), BRAÇLLIURE
 Efectes: NO SOBRELATAULA(x), NO BRAÇLLIURE
 AGAFAT(x)

BAIXAR(x)
 Prec.: AGAFAT(x)
 Efectes: NO AGAFAT(x)
 SOBRELATAULA(x), BRAÇLLIURE

Exercici 4. Planificació per agents de software*Part A:*

Definir els operadors que calguin per imprimir un fitxer en Unix, partint dels predicats: `es_fitxer`, `impres`.
Opcionalment també podeu fer servir el predicat `permis_lectura`.
Comprova si un fitxer executable es pot imprimir

Part B:

Ampliar el domini perquè no ens calgui explicitar els fitxers que tenim a l'estat inicial.
Això es pot fer definint una acció de recollida d'informació: `ls`. La definirem estàticament a partir del predicat: `dir_actual` (i opcionalment podeu definir `in_dir`).

Es definirà una acció `ls` per a cada directori. Com influència en l'eficiència del planificador el fet d'augmentar el nombre d'accions

Part C:

Ampliar el domini perquè no ens calgui donar informació sobre el directori actual. (Això és, definir la acció *estàtica* de recollida d'informació: `pwd`).

Part D:

Ampliar el domini per controlar que els fitxers comprimits no es poden imprimir. En cas de que estiguin comprimits, crear una acció per descomprimir-los.
Com queda el fitxer comprimit després d'imprimir-lo?

Part E:

Ampliar el domini per enviar un document per e-mail a un usuari del qual només coneixem el nom.
(Pista: usar l'acció de recollida d'informació `finger`).

Exercici 5. Softbots

Volem enviar un document a un usuari donat a través de la comanda Unix **write**. Aquesta comanda accepta dos paràmetres: el canal o terminal on està connectat l'usuari (identificat amb el seu *login*), i a continuació el text a enviar. Si l'usuari al qual se li vol enviar el document no està connectat, la comanda *write* fracassa. El document està en alguna part de l'arbre de directoris l'arrel del qual és el directori *home* de l'usuari que vol enviar el missatge, però no tenim la certesa absoluta. Proporciona una llista d'operadors per poder planificar la seqüència de comandes que cal realitzar en Unix per poder assolir el nostre objectiu.
(Comandes en Unix que us poden ser útils: *find*, *finger*).
Està clar que en aquest exercici heu d'usar accions de recollida d'informació. Quin tipus de tècnica de planificació s'haurà d'utilitzar per assegurar que el pla construït no fracassi?

Exercici 6. Un problema criptoaritmètic

Es tracta d'assignar un número (entre 0 i 9) a cada lletra de forma que:

- No poden haver dues lletres amb el mateix valor
- La suma de tots els dígitos ha de ser la que es mostra a continuació:

$$\begin{array}{r} \text{SEND} \\ + \text{MORE} \\ \hline \text{MONEY} \end{array}$$

Formular el problema com un problema de satisfacció de restriccions (CSP).

Una primera aproximació a la resolució del problema és considerar l'espai de cerca on a cada estat tenim un conjunt de lletres (variables) assignades, i un altre pendent d'assignar.

L'estat solució és aquell que té totes les lletres assignades i que satisfà les restriccions de l'enunciat.

Calculeu la dimensió de l'espai explorat fins el moment que es troba la solució.

Quines heurístiques podeu usar per millorar l'eficiència?

És possible formular el problema d'aquesta pràctica com un problema CSP binari?

Si creieu que sí, feu-ho. Si creieu que no, expliqueu per què.

Exercici 7. Scheduling.

Suposa que tens que dissenyar un programa per decidir a quin dia i a quina hora es realitzarà el proper claustre de la Universitat de Girona. Formula el problema, indicant el coneixement que et caldria per resoldre'l. Distingeix el que serien activitats, recursos, capacitats, operacions, duracions, i fins i tot preferències (no fer un claustre durant el mes d'agost, per exemple).